AD-A053 684 PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y F/G 9/4
INTERACTIVE DIGITAL RECEIVER SIMULATOR (IDRS) SYSTEM USER'S MAN--ETC(U)
MAR 78 R E BOZEK, J C BREDA, O P GUPTA F30602-76-C-0126
UNCLASSIFIED PAR-76-30 RADC-TR-78-44 NL

1 OF 2
AD
A053684

AD A053684

DDC FILE COPY

RADC-TR-78-44
User's Manual
March 1978

INTERACTIVE DIGITAL RECEIVER SIMULATOR (IDRS) SYSTEM USER'S MANUAL AND SOFTWARE DOCUMENTATION

Mr. Robert E. Bozek
Mr. Joseph C. Breda
Dr. Om P. Gupta
Mr. Roland H. Holman
Mr. Raymond J. Staron
Mr. Dennis D. White

Pattern Analysis and Recognition Corporation

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-44 has been reviewed and is approved for publication.

APPROVED: Richard S Jones

RICHARD S. JONES
Project Engineer

APPROVED: Fred I Diamond

FRED I. DIAMOND
Technical Director
Communications and Control Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

(14) PAR-76-30

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (18) RADC-TR-78-44	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) INTERACTIVE DIGITAL RECEIVER SIMULATOR (IDRS) SYSTEM USER'S MANUAL AND SOFTWARE DOCUMENTATION.		5. TYPE OF REPORT & PERIOD COVERED User's Manual
		6. PERFORMING ORG. REPORT NUMBER PAR Report No. 76-30
7. AUTHOR(s) (10) Mr. Robert E./Bozek, Mr. Joseph C./Breda, Dr. Om P./Gupta, Mr. Roland H./Holman, Mr. Raymond J./Staron, ~~Mr. Dennis D. White~~		8. CONTRACT OR GRANT NUMBER(s) (15) F30602-76-C-0126
9. PERFORMING ORGANIZATION NAME AND ADDRESS Pattern Analysis and Recognition Corporation 228 Liberty Plaza Rome NY 13440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F (17) 10 (16) 55561003
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCIT) Griffiss AFB NY 13441		12. REPORT DATE (11) Mar[redacted]78
		13. NUMBER OF PAGES 154
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED (12) 148p.
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Richard S. Jones (DCIT)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Signal Processing
Spectral Analysis
Histogram Quadrature Detection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the design and implementation of the Interactive Digital Receiver Simulator (IDRS). The IDRS is a computer program that is capable of processing predetected signals that have been converted to digital format and recorded. It allows a user to computer interactively produce (1) desired hard-copy patterns of signals for a detailed analysis of modulation characteristics and (2) processed signal segments for processing by pattern analysis/recognition programs for feature extraction. In essence, the IDRS program provides the user

390 101

Gee

with the capability of interactively creating a digital receiver to his specifications.

INTRODUCTION

This document describes the Interactive Digital Receiver Similulator (IDRS) System. The system was developed at PAR and is an augmentation of the Long Waveform System (LWS) developed at RADC.[1] Additional signal processing tools were added by PAR, some of which are: a digital filter, and the capability of software demodulation of predetection waveforms.

Section 1 of this report describes the system from a user's point of view. Plots are are included to show a typical interactive session.

Section 2 describes the software structure and program documentation from a programmer's standpoint. This section has been incorporated to aid in future software additions or modifications.

Two appendices are incorporated into this report. Appendix A is a technical discussion of the digital filter design techniques incorporated into IDRS. Appendix B describes a version of IDRS which was modified to meet the requirements of the Emitter Identification Program (EIP).

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

SECTION 1

(U) DEVELOPMENT OF SOFTWARE FOR DIGITAL PROCESSING OF PREDETECTION SIGNALS

In the Ft. Huachuca collections, and such tests conducted at PAR, wideband predetection signals were analog-recorded on magnetic tape. Selected analog recordings have been converted to digital format at PAR's analog-to-digital conversion (ADC) facility. This section discusses the design and use of software for computer processing of the resulting predetection digital signal recording. The predetection software, termed the Interactive Digital Receiver Simulator (IDRS), was developed as a task under contract #F19628-76-C-0002 and augments the capability of the RADC/PAR Waveform Analysis Facility. Under the current effort, several software additions have been made to IDRS.

IDRS allows a user/analyst to computer-interactively produce: (1) desired hardcopy patterns of signals for a detailed human analysis of modulation characteristics and (2) processed signal segments for input to the WPS for feature extraction. In so doing, the IDRS user interactively creates a digital receiver to his specification.

The discussion in this section of the IDRS design and use will illustrate the hardcopy pattern generation.

IDRS is realized as an augmentation to the Long Waveform System (LWS)[1] which had been previously developed as part of the RADC pattern recognition capability. The LWS architecture was deemed more suitable than that of the Waveform Processing System (WPS) for implementing predetection processing algorithms. The principle reason is that the LWS was designed for in-depth analysis of one or two waveforms. In contrast the WPS has capability for processing and feature extraction from multi-waveforms, but is inefficient when used for in-depth processing of a single long waveform. Also, much of the Long Waveform System is in FORTRAN and has provision for an expanded number of FORTRAN overlays. These factors made it possible for the predetection processing algorithms to be realized with much less programming effort than would have been necessary with the assembly language programming required for WPS.

IDRS made use of the existing LWS architecture for data storage, flexible data accessing, fast Fourier transformation, and display capabilities. These capabilities are discussed in Reference 1. The predetection algorithms are realized as overlays to the LWS. Some changes were also made to the LWS overlay management to more efficiently accomodate the predetection processing overlays. In addition, the system has been modified to run under the RSX-11D operating system. This was accomplished to permit access to all 80K of core memory*, and allow usage in a multiple user environment.

---

* The DOS FORTRAN system restricts usage to 28K.

The remainder of this section discusses the selection of predetection signal processing algorithms, their realization, and use.

1.1. REQUIRED DIGITAL PROCESSING CAPABILITY

Figure 1-1 illustrates the signal processing capability that was desired and then realized with IDRS. The inputs to the system are digitized predetection (on a carrier) waveforms. The first function required of the system is to aid the analyst in viewing the signal to select a desired portion for in-depth analysis. In this process, termed segmentation, the signal is displayed both as a function of time and frequency.

Next, using a power spectrum display, the analyst selects from his point of view the most desirable preselection filter. The analyst specifies the filter in terms of center frequency, bandwidth, and roll-off. He selects the filter to best reject noise and interference and also as a first step in achieving a greater frequency resolution in displays of power spectra.

Frequency resolution in spectral analysis is related to the duration of a signal segment that is Fourier transformed. For digital spectral analysis, the time interval is fixed by the sampling rate and size of the Fast Fourier Transform (FFT). The options for increasing frequency resolution in digital spectral analysis are to increase the size of the FFT or to decrease the sample rate. Presently, the IDRS has a maximum size FFT of 1024 complex points. To increase the size FFT in IDRS would require significant program architectural changes. Therefore, the selected approach to increased frequency resolution is to decrease the sampling rate. To accomplish this the analyst selects a portion of the spectrum or frequency range over which he wishes to achieve high resolution analysis. He then specifies a predetection filter that selects signal frequency components only within this frequency range of interest, and proceeds to filter the signal. Since the filtered signal is now further band limited, the analyst can resample to achieve a lower sampling rate. The resampling can be accomplished by simply retaining every n^{th} sample. The analyst must use judgment in selecting the resampling rate in order not to introduce aliasing. Figure 1-1 shows the predetection filtering and resampling as an iterative process for achieving increased frequency resolution or "zoom-in". If the analyst is not satisfied with the frequency resolution after viewing a power spectrum display resulting from a zoom-in, he may elect to accomplish a greater zoom-in.

Figure 1-1 also indicates the desired capability for AM and FM demodulation. It is noted that the analyst needs to have control over both the predetection and postdetection filter properties. The operator then has the option of viewing demodulated waveforms or their power spectra.

In calculating Power Spectral Density (PSD) plots, the analyst needs options for waveform weighting, averaging of raw PSDs, and linear and logarithmic power scale. The options for waveform weighting are required where the PSD varies over a wide power range as a function of frequency. By selecting the waveform weighting the analysts can control the shape of the

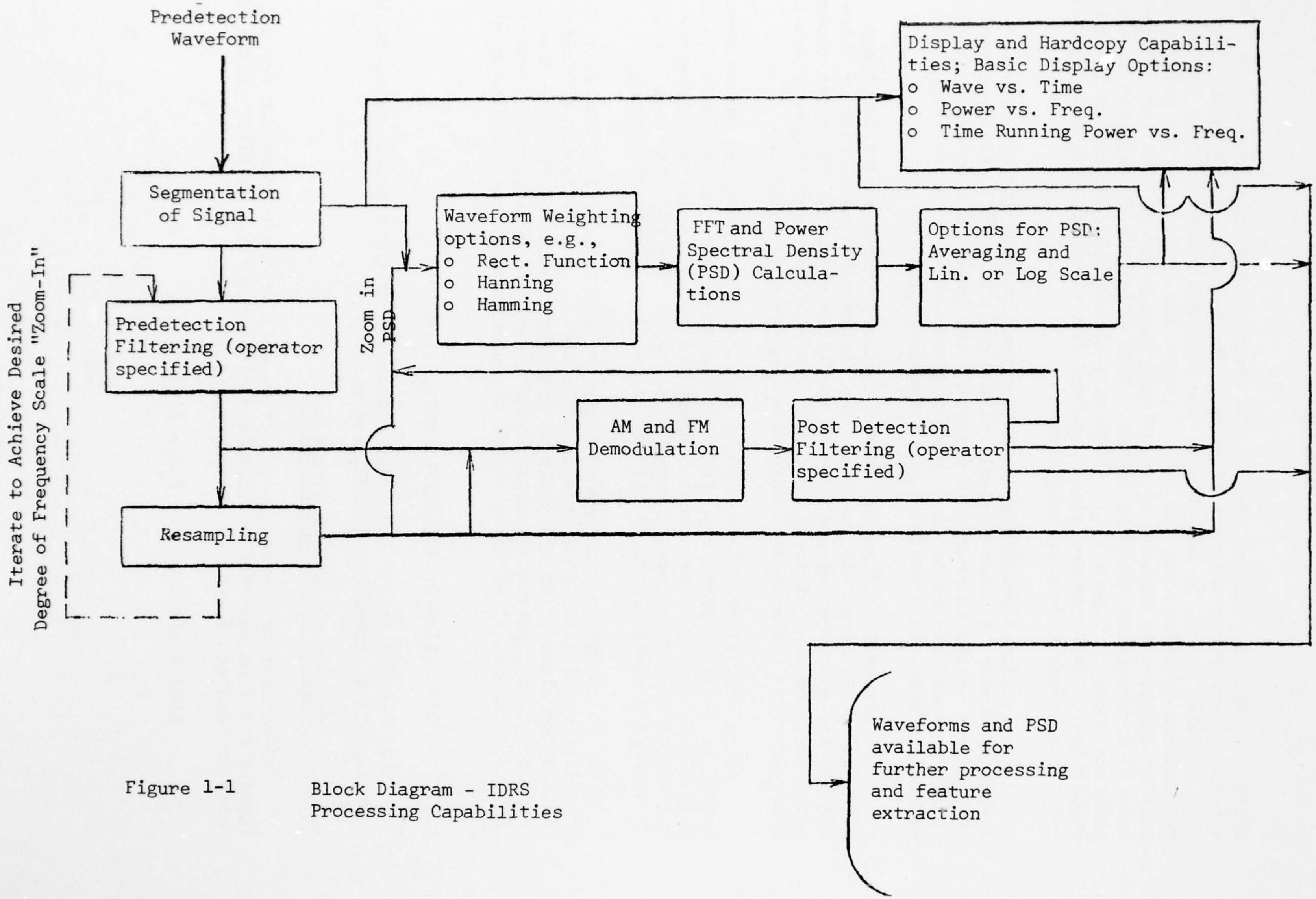


Figure 1-1 Block Diagram - IDRS Processing Capabilities

effective analysis filters used in the spectral analysis process (the FFT is a band of contiguous analysis filters). By choosing an analysis filter shape with low side lobes, he can obtain more reliable spectral power estimates at frequencies of relatively low PSD. Without low side lobes the power estimate will be affected by regions of relatively high power at side lobe frequencies near to a particular analysis filter. This effect is illustrated in the spectrum shown in Figure 1-2. (Figures 1-11 and 1-12 in Section 1-4 show respectively IDRS-produced signal spectra for a Hanning-weighted signal versus the signal without Hanning weighting.)

By selecting a weighting or window function with low side lobes in the frequency domain, a wider analysis filter main lobe will result. Hence, if the analyst wishes to precisely determine the location of a frequency domain spike, he would desire to use a different weighting function than if he desired to accurately estimate power spectral density.

IDRS presently offers the user a choice of three window functions: rectangular, Hanning, and Hamming. The mathematical properties of these windows are discussed extensively in the literature.[2] The three windows allow the user a significant choice in trade-off between frequency domain side lobe reduction and main lobe resolution in spectrum analysis.

Since many of the signals to be analyzed are made up of random components, at least in part, any single or raw power spectrum estimate (periodogram)[2] is likely to be a poor representative of the average properties of the signal.[2] Therefore, it was necessary to give the IDRS user the option to average a number (1 to 99) of raw PSD estimates or periodograms.

The third user option needed for power spectral density analysis was that of selection of scale in displaying PSD plots. In addition to a linear power scale a logarithmic (or dB) scale is required because of the wide dynamic range of power spectrum associated with signals to be analyzed. Thus, the IDRS user can select either a linear or logarithmic scale for displaying PSD plots.

1.2. DISCUSSION OF ALGORITHMS USED IN IDRS

This section provides a brief discussion of the approach taken to implement important algorithms in IDRS software.

1.2.1. Digital Quadrature Detector

The concept of quadrature detection will be first presented for analog signals and then extended to include digital signals. We first define any radio or bandpass signal by the expression,

$$x(t) = a(t) \cos [2\pi f_c t + \emptyset(t)]$$

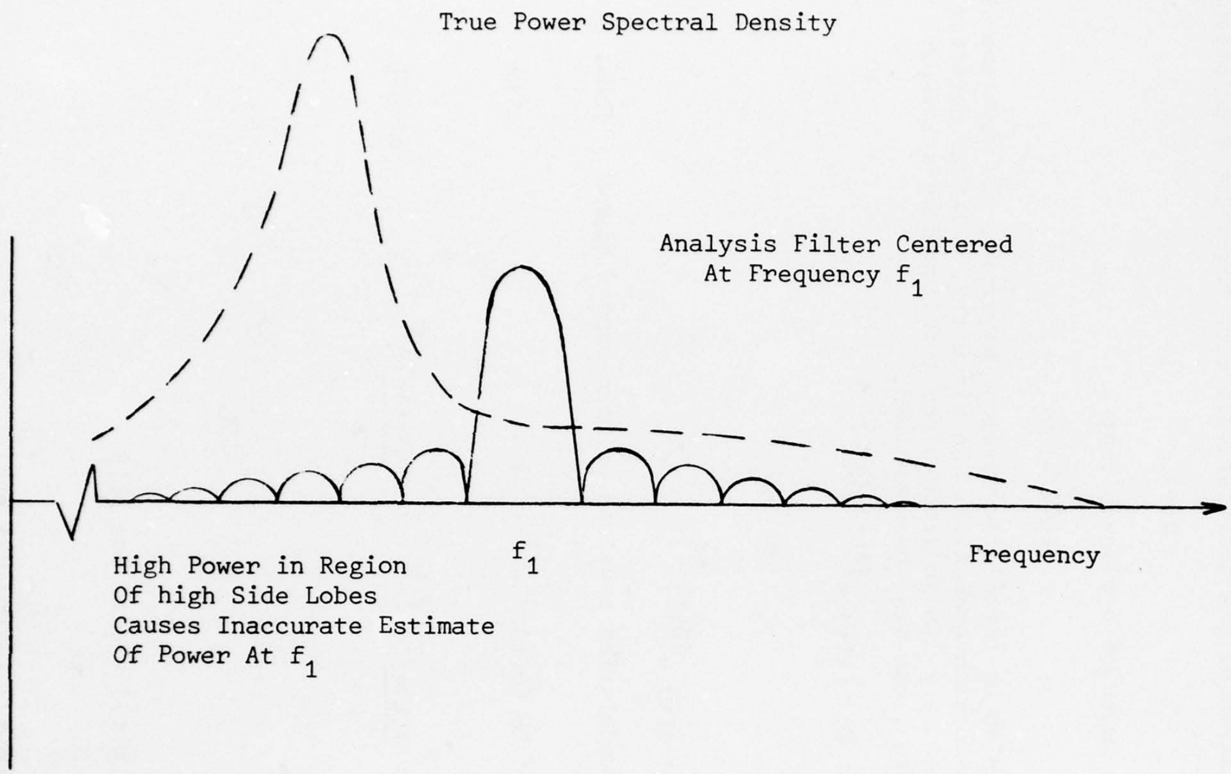


Figure 1-2 Inaccurate Power Spectral Density Estimation Due To Analysis Filter Sidelobe Leakage

where, t is the time variable, f_c is the carrier frequency, a(t) is the amplitude modulation, and $\phi(t)$ is the phase modulation. We also define the frequency modulation as the time derivative of the phase modulation, that is,

$$\dot{\phi}(t).$$

Furthermore, we restrict a(t) such that

$$a(t) \geq 0.$$

Thus, a(t) can also be termed the envelope waveform, the waveform that results from a linear detection operation. In circuitry, a linear detector is commonly achieved by a linear diode circuit followed by a low-pass filter. Our bandpass signal can also be expressed,

$$x(t) = \text{Re}\ [a(t)\ e^{j\phi(t)} e^{j2\pi f_c t}]$$

where $j \equiv \sqrt{-1}$.

We define,

$$m(t) = a(t)\ e^{j\phi(t)}$$

as the complex modulation envelope or base-banded signal. Thus,

$$x(t) = \text{Re}\ [m(t) e^{j2\pi f_c t}] \qquad (1)$$

$$= \frac{m(t) e^{j2\pi f_c t}}{2} + \frac{m^*(t) e^{-j2\pi f_c t}}{2} \qquad (2)$$

We next define,

$$m_I(t) = \text{Re}[m(t)] = a(t) \cos \phi(t)$$

$$m_Q(t) = \text{Im}[m(t)] = a(t) \sin \phi(t)$$

where $m_I(t)$ and $m_Q(t)$ are known respectively as the in-phase and quadrature-phase waveforms. The modulation signals can now be expressed as a function of $m_I(t)$ and $m_Q(t)$

$$a(t) = \sqrt{m_I^2(t) + m_Q^2(t)}$$

$$\phi(t) = \tan^{-1}\left[\frac{m_Q(t)}{m_I(t)}\right]$$

Thus, if we can extract the in-phase and quadrature-phase waveforms from a band-pass signal, the modulation waveforms can also be determined. The process of extraction of the quadrature waveforms is termed quadrature detection.

It is useful in presenting the signal processing approach to quadrature detection to express signals in the frequency domain. We define X(f) as the Fourier transform of signal x(t), i.e.,

$$x(t) \overset{F}{\longleftrightarrow} X(f)$$

where f is the frequency variable. We now make use of two well-known theorems[3] involving Fourier transform pairs. The first, known as the "shifting" theorem, states that if

$$y(t) \overset{F}{\longleftrightarrow} Y(f)$$

then

$$y(t)\, e^{-j2\pi f_s t} \overset{F}{\longleftrightarrow} Y(f + f_s) \qquad (5)$$

We will call f_s the shift frequency. The second theorem, the complex conjugate theorem, states that if

$$y(t) \overset{F}{\longleftrightarrow} Y(f)$$

$$g(t) \overset{F}{\longleftrightarrow} G(f)$$

---

* This denotes that x(t) and X(f) are Fourier transform pairs.

and

$$g(t) = y^*(t),$$

where $y^*(t)$ is the complex conjugate of $y(t)$, then

$$G(f) = Y^*(-f) \qquad (6)$$

Applying (5) and (6) to equation (2) we can express the spectrum of our signal as

$$X(f) = \frac{M(f - f_c)}{2} + \frac{M^*(-f-f_c)}{2} \qquad (7)$$

where $M(f)$ is the Fourier transform of the complex modulation envelope, $m(t)$. If the spectrum of $m(t)$ is as shown in Figure 1-3a, then by (7), the spectrum of $x(t)$ is as shown in Figure 1-3b.

The schematic shown in Figure 1-4a shows the extraction of the complex modulation waveform by applying the Fourier shifting theorem followed by a low-pass filtering operation. The signal is multiplied by

$$e^{-j2\pi f_c t},$$

resulting in a complex signal. The spectrum of this complex signal is shown in Figure 1-3c. The shifted spectrum consists of a base banded component and a double frequency component. It is noted that the base-banded component is proportional to the complex modulation envelope spectrum. Thus, the complex modulation envelope can be extracted by a filtering operation that rejects the double frequency term and a gain adjustment. An appropriate low-pass filter (LPF) response is shown in Figure 1-3d. Figure 1-3e shows that the extracted signal spectrum is proportional to that of the desired complex modulation envelope. If the pass-band gain of the LPF is one, then the complex modulation signal is obtained by weighting the filter output by a factor of two.

In practice, the complex modulation envelope is not obtained directly, but instead the quadrature waveforms are first extracted. Figure 1-4b shows

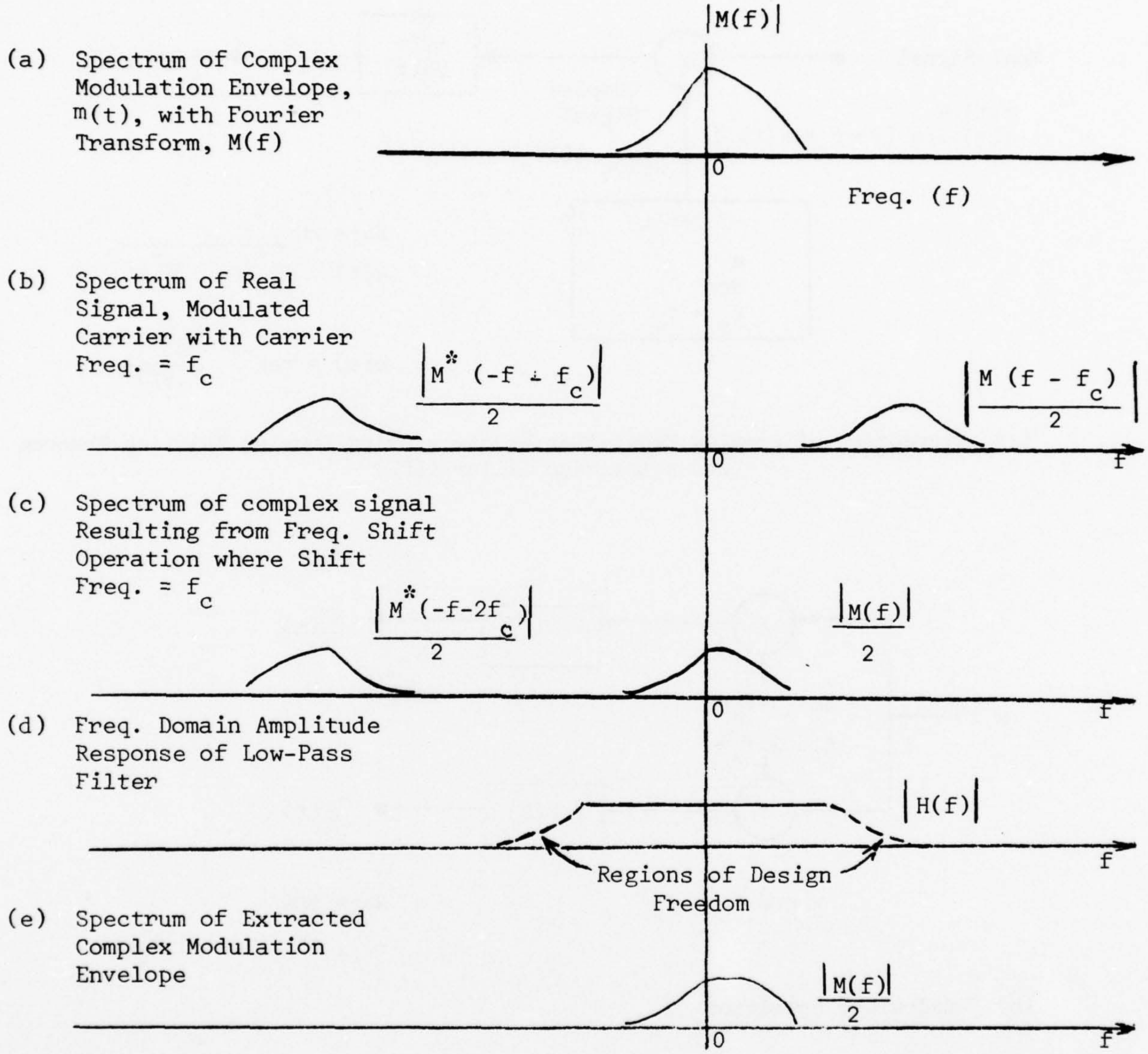
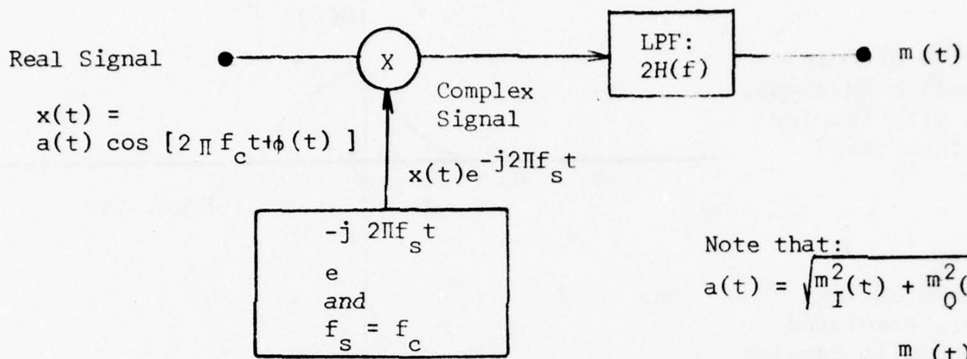


Figure 1-3 Spectra Associated with Steps In Extraction of Complex Modulation Envelope

Figure 1-4 Schematics of Complex Modulation Envelope Extraction

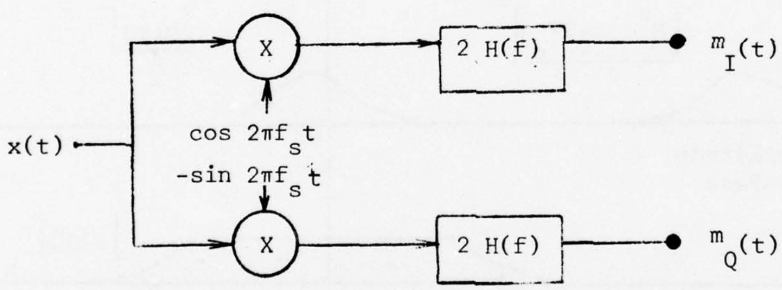


Note that:

$$a(t) = \sqrt{m_I^2(t) + m_Q^2(t)}$$

$$\phi(t) = \tan^{-1} \frac{m_Q(t)}{m_I(t)}$$

(a) Extraction of Complex Modulation Waveform Using Fourier Shifting Theorem and a Low Pass Filter (LPF)



where $f_s = f_c$

Note that:

$$m(t) = m_I(t) + j\, m_Q(t)$$

(b) Quadrature Detection

KEY

t	=	time
f_c	=	carrier freq.
$a(t)$	=	envelope modulation
$\phi(t)$	=	angle modulation
j	=	$\sqrt{-1}$
f_s	=	shift freq.
$m(t)$	=	complex modulation envelope
	=	$a(t)e^{j\phi(t)}$
$H(f)$	=	Low Pass Filter Freq. Domain Response
$m_I(t)$	=	Re $[m(t)]$ = In-Phase Modulation Waveform
$m_Q(t)$	=	I_M $[m(t)]$ = Quadrature - Phase Modulation Waveform

a schematic of a quadrature detector. The signal is first split into two paths, and the path signals multiplied respectively by the real and imaginary parts of $e^{-j2\pi f_c t}$. Each of the resulting signals is then low-pass filtered. The resulting signals, given the proper gain, are identically the in-phase, $m_I(t)$ and quadrature-phase, $m_Q(t)$ waveforms.

The quadrature detector has the advantage that all processing is accomplished with real signals. Quadrature detectors have been realized in analog hardware for applications such as coherent radar signal processing.

In general, the carrier frequency, f_c is not known precisely, but is estimated. The error in selection of the shift frequency, f_s is defined as

$$\Delta f = f_c - f_s$$

This error affects the extracted complex modulation envelope, which is now symbolized as

$$\hat{m}(t) = \hat{m}_I(t) + j\,\hat{m}_Q(t)$$

and where $\hat{m}_I(t)$ and $\hat{m}_Q(t)$ are the quadrature waveforms extracted by a quadrature detector. The relationship between m(t) and $\hat{m}(t)$ can be expressed as follows:

$$\hat{m}(t) = a(t)\, e^{j\emptyset(t) + 2\pi \Delta ft}$$

$$= m(t)\, e^{j2\pi \Delta ft}$$

Hence, the extracted envelope waveform is identically a(t) and the extracted angle modulation is, $\emptyset(t) + 2\pi \Delta ft$. However, we are more interested in the extracted frequency modulation (FM) waveform than the phase modulation. The extracted FM is expressed as,

$$\hat{\dot{\emptyset}}(t) = \frac{d(\emptyset(t) + 2\pi \Delta ft)}{dt}$$

$$= \dot{\emptyset}(t) + 2\pi \Delta f$$

Thus, the error in selecting the frequency shift results in a dc level equal to $2\pi \Delta f$. This dc level can easily be removed from the extracted FM waveform.

The processing of a digital band-pass signal resulting from the process of digitizing a band-pass signal, is in many ways analogous to the analog process discussed above. The symbols t and f are now respectively discrete time and frequency. The Fourier transform theory is replaced by the discrete Fourier transform (DFT) and its analogous theorems.[3] The significant conceptual difference between digital and analog processing is that spectra of digital signals are periodic. Figure 1-5 shows the periodic spectra associated with digital extraction of the complex modulation envelope. In computer developed displays of the complex modulation envelope, generally only that portion of the frequency axis between $-\frac{f_p}{2}$ and $\frac{f_p}{2}$, or less, is plotted.*

In summary, the IDRS approach for extraction of the complex modulation envelope waveform for a digital signal,

$$x(t) = a(t) \cos[2\pi f_c t + \phi(t)]$$

is to first obtain a display of the power spectrum of x(t), that is

$$| X(f) |^2$$

The user/analyst then makes use of the power spectrum (or averaged power spectrum) to select a shift frequency, f_s and to specify the low-pass filter bandwidth and shape parameters.

The signal is then multiplied by both $\cos 2\pi f_s t$ and $\sin 2\pi f_s t$. Each of these resulting signals is then low-pass filtered with a filter meeting the user's specifications. The filtered signals are respectively the real and imaginary parts of the extracted digital complex modulation envelope.

The computation of products of a band-pass signal with $\cos 2\pi f_s t$ and $\sin 2\pi f_s t$ can be made rather efficient if the shift frequency, f_s is made equal to a multiple of DFT frequency sampling interval. If this is the case, the values of the time series, $\cos 2\pi f_s t$ and $\sin 2\pi f_s t$ can be obtained directly from a computer table associated with the FFT routine.

An alternative processing technique for extraction of the complex modulation envelope of a digital band-pass signal is the application of the

---

* $f_p \equiv$ Sampling frequency

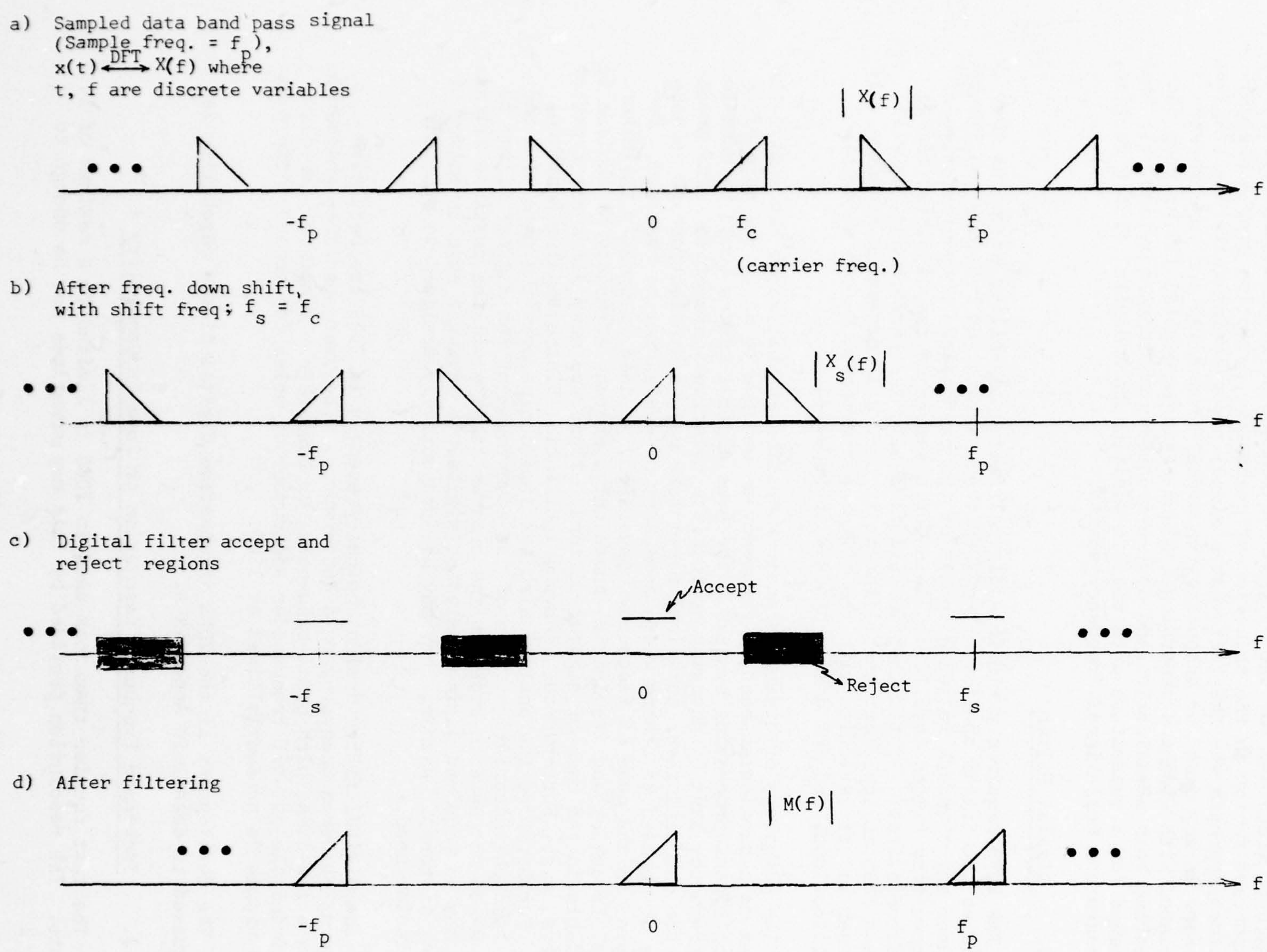


Figure 1-5 Spectra of Digital Signals Associated With Extraction Of Digital Complex Modulation Envelope

Hilbert transform.[4,5] The quadrature detection approach was selected over the Hilbert transform for use by IDRS for reasons of computational efficiency. An important factor in the selection of quadrature detection over a Hilbert transform approach was the fact that a signal predetecting filtering step was necessary in any case for enhancing the signal from noise and interference associated with wideband recording. Since a single digital filter step can accomplish both preselection and double-frequency content rejection, all that is needed for a quadrature detector, in addition to a digital filter routine, is quadrature real-signal frequency shifts.

1.2.2. Digital Filter

The IDRS requires a flexible digital filter capability where the user specifies the filter on-line. The IDRS then designs and realizes the filter. The user may then ask to see displays of the performance characteristics of the resulting filter (see Figure 1-16 for a sample of the displayed filter characteristics). If the user is satisfied with the performance, he will command IDRS to implement the filter in an over-all processing sequence. If the user is not satisfied with the filter performance, he can alter his specifications, and the design process is repeated.

The advantage of digital signal processing is that we do not commit the system to a particular band of frequency as we have to do in the case of analog signal processing systems. The same digital system could be used for any frequency band by choosing appropriate numerical parameters. The procedure for choosing these numerical parameters may be straightforward in some cases as opposed to others where some sort of design may be required. For example, in the case of frequency conversion, we need to specify the mixer signal frequency and its initial phase only; whereas a variety of choices is available in the case of digital filters. There are many different types of filters, e.g., Butterworth, Chebyshev, Elliptic, Finite Impulse Response, etc. Furthermore, the analyst also has some choice over the pass band and stop band characteristics. One of the advantages of the digital filter is that we do not have to store in the program library all the possible filters that may be required during analysis (similar to having a huge inventory of analog filters). Rather, they may be interactively designed to suit any particular need.

The digital filter design choices presently in IDRS are recursive filters, either a Chebyshev or a Butterworth. The user specifies whether he wants a low-pass, high-pass, band-pass, or band-reject response. He also specifies the cut-off frequencies and number of poles (up to 18). The pass-band ripple is presently fixed at 1 dB.

The development of the IDRS interactive digital filter capability is discussed in detail in Appendix A.

1.2.3. Increased Frequency Resolution or Zoom-In Capability

The fast Fourier transform used in IDRS is limited to a maximum of 1024 points. The resolution provided by this may sometimes not be enough to

display the fine grain structure of the spectrum. One way to increase the resolution is to increase the FFT capability to more than 1024 points. However, this upper limit of 1024 points has been dictated by the memory space available in computer hardware. An alternate scheme is implemented in IDRS that increases the resolution to any order. The block diagram of this scheme is shown in Figure 1-6. First, the band of frequency around which the magnified resolution is to be obtained is translated to zero frequency. Then a low-pass filter is used to curtail the bandwidth of the signal to a value for which the magnified resolution is required. Now, since the bandwidth of the low-pass signal has been reduced, it can be resampled. A 1024 point FFT of this resampled wave will provide an increased resolution of the desired spectrum band. Of course, this scheme assumes that more than 1024 points are available in the original waveform so that it can be resampled to obtain 1024 points. There is no way to increase the resolution if the original waveform did not contain enough data. An example of zoom-in spectrum analysis is given in Section 4.2. Figure 1-21 in section 4-2 is the power spectrum display of a certain band of a waveform spectrum using the "detail" option (to show any segment of a plot on an enlarged scale), whereas Figure 1-20 is the display of the same band using the zoom-in capability. A resampling rate of 10 to 1 was used for this option and hence a resolution 10 times greater can be expected in Figure 1-20 as compared to that in Figure 1-21.

1.2.4. Demodulation

The IDRS extracts amplitude modulation (AM), phase modulation (PM) and frequency modulation (FM) waveforms from the quadrature waveforms:

$$a(t) = \sqrt{m_I^2(t) + m_Q^2(t)}$$

$$\phi(t) = \tan^{-1} \frac{m_Q(t)}{m_I(t)}$$

$$\dot{\phi}(t) = \frac{\phi(t) - \phi(t - \Delta t)}{\Delta t}$$

where

t = time index

Δt = sampling interval

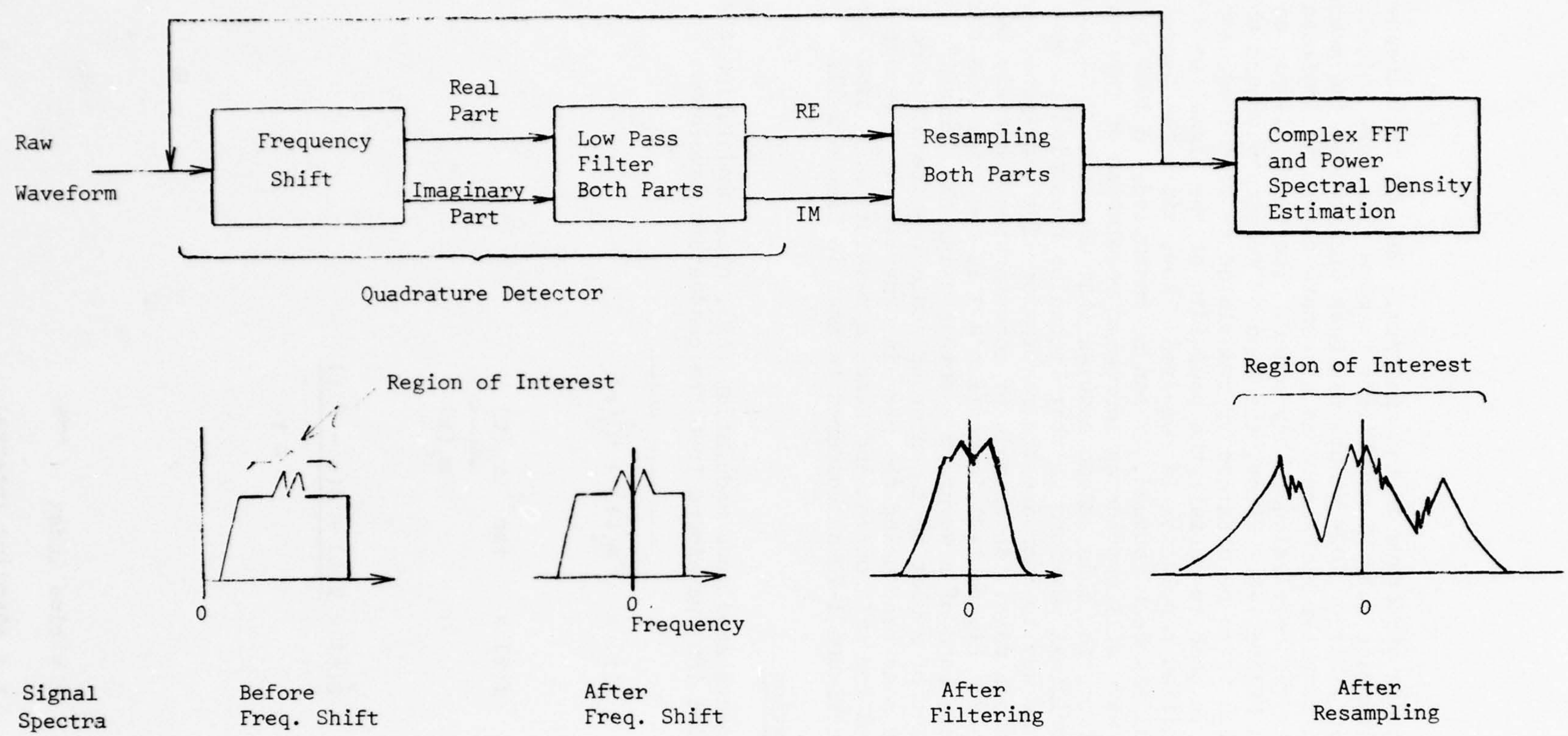


Figure 1-6 Block-Diagram of Power Spectrum: Zoom-In Scheme

1.3. IDRS SYSTEM SOFTWARE/HARDWARE

1.3.1. Software

The IDRS has been implemented on the PDP 11/45 and runs under the RSX-11D Operating System (RSX) with Tektronix 4014-1 graphic display and 4631 hardcopy unit. The IDRS is written using linked overlays that are retrieved from an RK05 Cartridge disk and executed only when needed. The main program and subroutines are core resident and control overlay swapping. The overlays operate independently from the main program but share the same common area for communication. The main program is written in such a way that additional overlays can be added with minimum effort.

Data retrieval and display routines are written to achieve maximum speed and efficiency. The system uses a combination of both FORTRAN and Macro Assembly Language routines to take advantage of the best of each. By using the assembly language for I/O routines, a great amount of FORTRAN overhead is eliminated. FORTRAN is used when possible for ease and speed in programming new additions.

The signal processing capabilities of IDRS are shown in the block diagram of Figure 1-7. This block diagram also illustrates how the user, by setting proper logic switches, can interactively achieve the waveform-to-waveform or waveform-to-power-spectrum transformation he chooses to display. The basic architecture of IDRS is such as to allow the user to make, in the simplest and most straightforward way, requests to programs operating on the PDP 11/45 computer. These requests, which can be considered much like the function key requests of a hand calculator, are entered into the computer via the terminal keyboard by one or two character commands. These character commands set flags in the system software which act as logic switches. Figure 1-8 shows a computer-displayed menu of commands available to the user.

For example, if the user wishes to take the power spectrum of the filtered raw data, he would first turn on the filter logic by giving a character command, then he would choose the power spectrum by giving another character command. The system would retrieve the raw data; skip over the rectify block (since the logic switch was not selected); then pass through the filter block to filter the data and skip over all the other transformation blocks except the FFT block to display the results.

The user may interactively design up to 18 pole Chebyshev or Butterworth digital filters. The filter can be of low-pass, high-pass, band-pass, or band-elimination type. He may also display the designed filter's characteristics such as impulse response, amplitude response, phase response, group delay, and step response for examination. The filter, once designed, remains active until it is redesigned. Of course, a new filter must be designed when starting cold on the system.

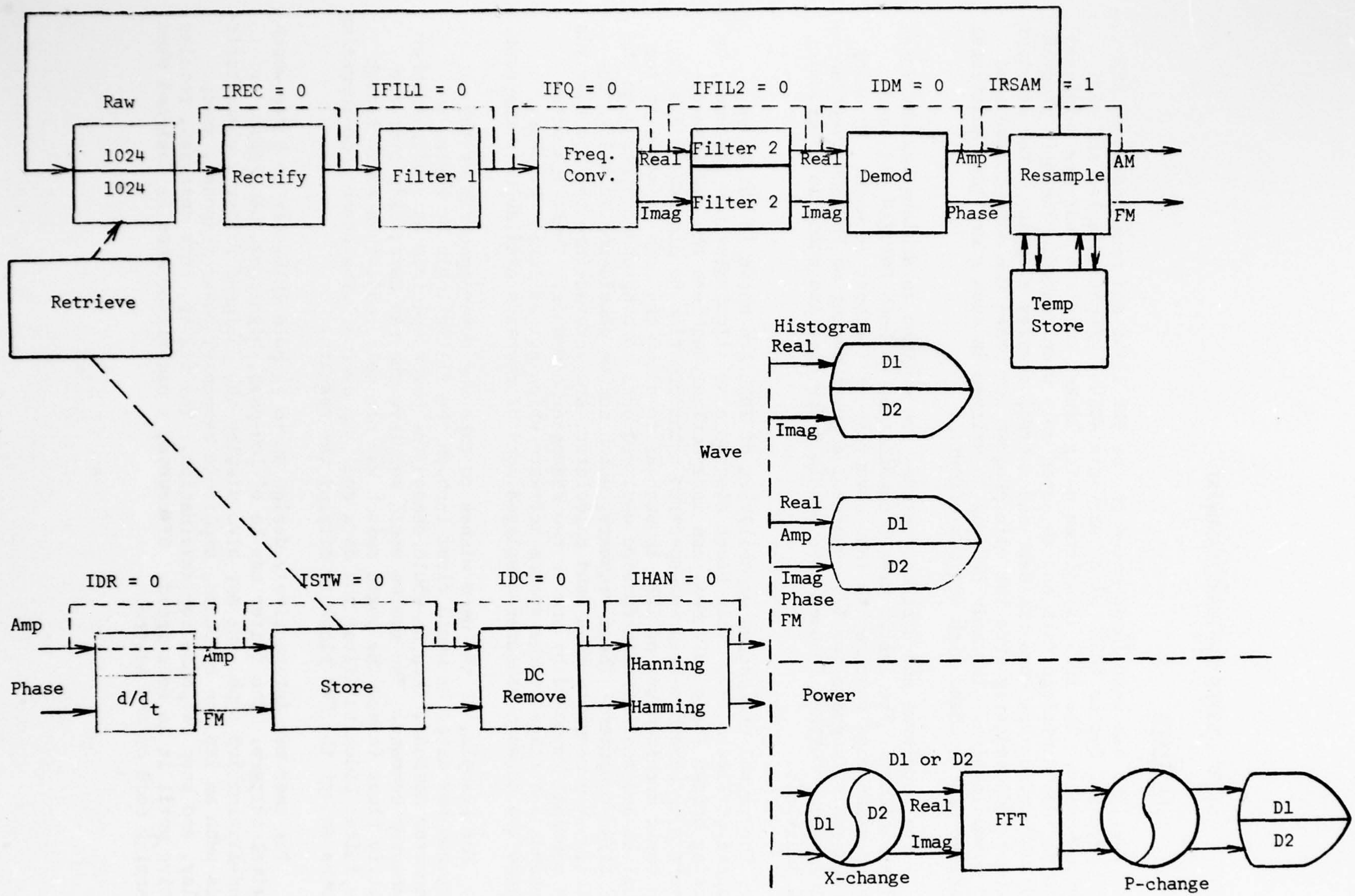


Figure 1-7 Functional Block Diagram of IDRS

```
R-RESET PARAMETERS
     RL-RESET # LINES
     RP-RESET # PTS/LINE
     RE-RESET SKIP POINTS (RESAMPLING)
     RO-RESET OVERLAP
     RS-RESET SKIP POINTS

    M-MENU

    LL-LOCAL LINE SCALE
    LP-LOCAL PAGE SCALE
    GW-GLOBAL WAVE SCALE

    W-DISPLAY WAVE
    P-DISPLAY POWER SPECTRUM
    S-DISPLAY SAME DATA

    X-TEXT (PAGE)

    T-INPUT NEW TIME
    T+-ADVANCE AHEAD TIME
    T--ADVANCE BACK TIME

    N OR NP-NEXT PAGE
    WN-WAVEFORM NEXT
    PN-POWER NEXT

    L--PREVIOUS LINE (BACK 1 LINE)
    L+-NEXT LINE (AHEAD 1 LINE)

    DP-DETAILS POINTS (ITH.JTH)
    DL-DETAILS LINE (EXPLODE 1 LINE)
    DO-DETAILS OFF

    DI-DATA ON DISK
    TI-TAPE ROLLIN
    TO-TAPE ROLLOUT
    TD-TAPE DUMP (PAGE)
    TW-WPS FORMATTED TAPE
    TP-TAPE PLAYBACK (PAGE)

    PA-POWER SPECTRUM AVERAGE
    PL-LOG OF POWER
    PP-POWER PROFILE
    PH-HANNING/HAMMING WEIGHTING

    DM-DEMODULATION ON
    FQ-FREQ  CONV
    DQ-PHASE/DT
    FI-FILTER INIT
    FN-FILTER NO INIT
    WR-WAVEFORM RECTIFICATION

    MO-DEMODULATION OFF
    QO-FREQ  CONV  OFF
    HO-WEIGHTING OFF
    WO-WAVE OPTIONS OFF
    PO-POWER OPTIONS OFF
    FO-FILTER OFF
    SO-STORE MODE OFF
    O-ALL OPTIONS OFF

SELECT NEXT OPTION

SW-STORE MODE (WAVE)
SP-STORE MODE (POWER)

D1-DISPLAY MODE 1
D2-DISPLAY MODE 2

FD-FILTER DESIGN

K-KILL RETURN TO MONITOR

RT-RETRANSFORM (ALL OFF - STORE OFF)
DC-DC REMOVAL
H1-1D HISTOGRAM
H2-2D HISTOGRAM
```

Figure 1-8 Menu of Commands

In general, the user can turn "on" or "off" any of the logic switches shown in Figure 1-7 to achieve the desired display, and once these transformed waves have been displayed, they are put into temporary storage by the system. This allows the user to do a limited amount of retransformation of a transformed wave by setting other logic switches.

The following digital processing capabilities are available in IDRS: rectification, filtering, frequency conversion, demodulation, derivative of phase (FM). One may display any of these processed waveforms. For any waveform or processed waveform that can be displayed on the system, the power spectrum, average power spectrum, or the log (with dB scaling) of any of them can also be displayed. DC removal, Hanning, or Hamming weighting can be performed on any waveform prior to power spectrum computations. Frequency bands can be zoomed in on as desired. The power spectrum need not be restricted to just real data. The system senses when complex power must be taken and automatically informs the user of complete system status. The system has pseudo 3-D capability for displaying power spectrums (or waveforms) plus the capability to display power profiles over time. Histogram routines are available as an aid in analysis. A one-dimensional histogram plots frequency of occurrence versus waveform amplitude. Input data may be either waveforms or power spectra. A two-dimensional histogram capable of plotting frequency of occurrence versus waveform DC value versus waveform RMS value may be run apart from IDRS. Auto and cross-correlation routines are also available outside of IDRS. In the future, it is planned to incoporate these routines into the IDRS system.

IDRS will allow the user to interactively display on the storage tube for review, edit, or hardcopy any number of lines with any number of waveform points per line (up to 2048). The user can display selected lines and/or portions of a line on an expanded scale through a "detailed" option.

The waveform is stored on a mass storage disk (RP04) with a reference beginning time (hour, min, sec). This allows the data to be randomly retrieved from the disk starting at any specified time. Once a display is up, the user can go either to the next page, previous line, next line, plus or minus X points, or select a totally new time. Those options are valid independent of the type of display.

The display ordinate, that is waveform amplitude or spectrum power, may be scaled to local line, local page, or global wave. Scaling may be changed any time. Text information may be added to the display. The data may be rolled from magtape to the disk or stored from the disk to magtape under a convenient format.

The system will allow the user to dump a page of displayed data along with headers to magtape. This capability allows the user to not only have a paper hardcopy of the display but also a magtape file copy. These magtape page files can be selectively played back at a later time for further manipulation of data or feature extraction.

1.3.2. Hardware Environment

The IDRS has been implemented on the PDP 11/45 computer utilizing the RSX operating system. Programs are written and debugged in a multi-user computer environment and stored on a RK05 disk cartridge in a system loadable format (i.e., executable code). I/O operations on the 4014-1 storage tube terminal are performed by a direct link to the 11/45.

Data is stored on an RP04 disk pack having a 40 million word capacity. It is possible to access the disk pack independently of IDRS software. Thus, the user can create a data file on the disk pack from a digital data source such as magtape which can be accessed by the Display and Analysis Subsystem.

The Tektronix 4014-1 Storage Tube has a display surface with 4096 addressable points on the horizontal axis and 3040 addressable points on the vertical axis. The device provides full ASCII capability for input and output.

The Tektronix 4631 hardcopy unit produces an exact copy of the current 4014-1 display. The unit can be triggered manually or through software.

1.4. SAMPLE RESULTS OF DIGITAL PROCESSING OF A PREDETECTION SIGNAL

This section takes the reader through a typical IDRS processing sequence. Hardcopy displays for each step are presented and discussed.

The discussion begins with a sample of a digitized signal stored on the computer system disk. The signal is a predetection recording of an RT-524 transmission, that was originally analog recorded at a 100 kHz carrier frequency. The analog tape recording was played-back at 1/16th of the recording speed and digitized at a rate of 20K samples/sec. Thus, the effective sampling rate was 320K samples/sec.

1.4.1. Plot of Predetection Waveform

The user requests a display of a portion of the stored digital signal. Figure 1-9 shows a steady-state portion of the RT-524 predetection recorded waveform.

The computer-produced annotation at the top of Figure 1-9 indicates to the user the processing operations that have been selected to produce the displayed result. "WP" indicates that a waveform is plotted. "LP" indicates that the waveform amplitude scaling is local page, that is the scaling is set based on the maximum waveform value appearing on the page. The user also has the option for local line (LL) scaling where each line of a display is scaled according to the maximum waveform value contained within the line of data. "C1" indicates that the data was retrieved from the data disk Channel 1 file. "REAL" indicates that the real part of the waveform is plotted (in this example the waveform itself is real). "SW" indicates that the IDRS used a

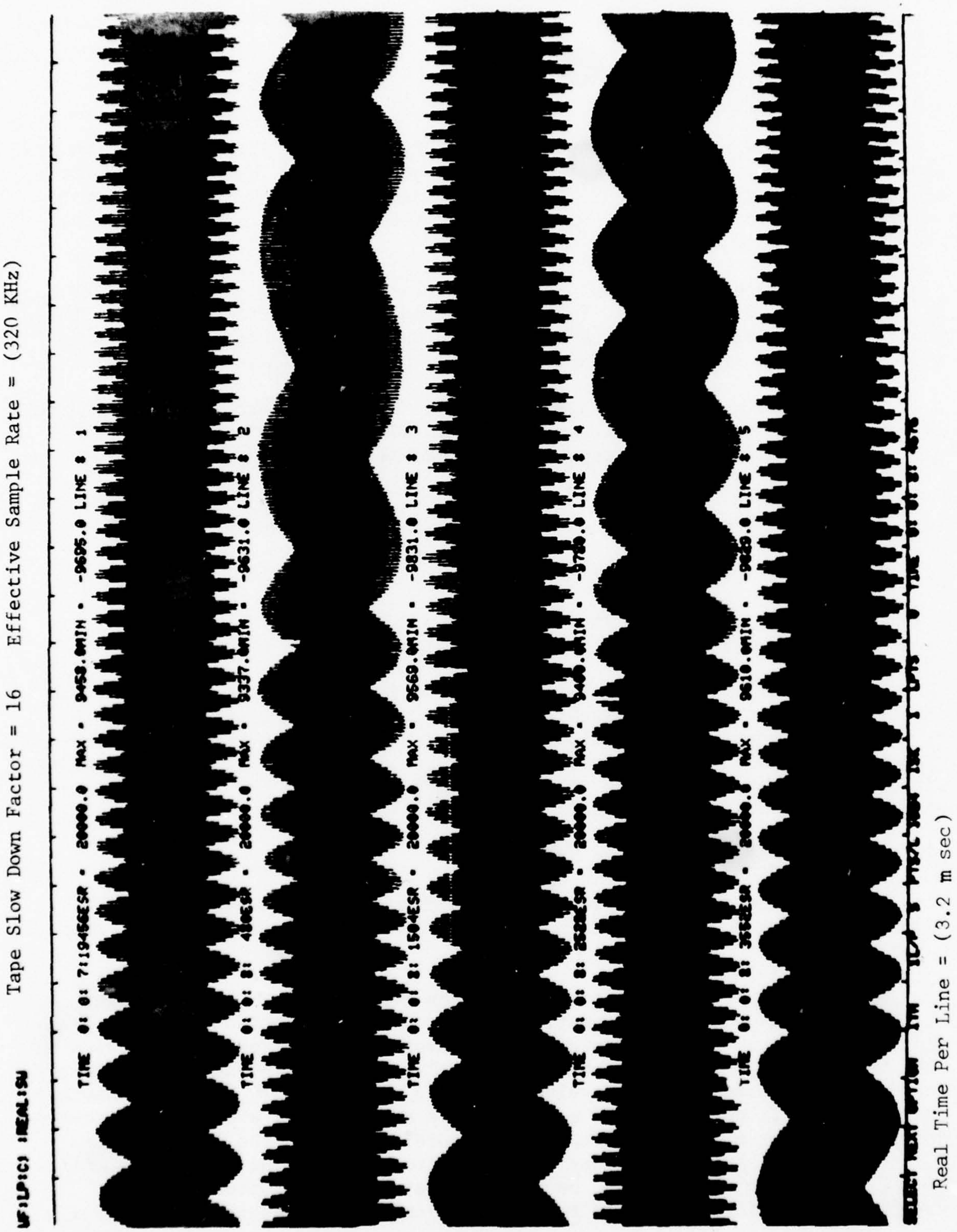


Figure 1-9 Display of a Recorded RT524 Predetection Signal

stored waveform mode, which is not significant for interpretation of the displayed result.

Referring to the computer-produced anotation at the bottom of the display, "SELECT NEXT OPTION" indicates to the interactive user that the system is ready for his next command. The "Ith" along with a "Jth" option allow the operator to select a waveform plot starting and stopping points for plotting a portion of a line of data with increased time scale. In Figure 1-9 the user elected to plot each entire line of data (e.g. I=1). "L/P 5" indicates that the user has elected to have displayed 5 lines of data per page. The user has the option of having from 1 to 99 lines displayed per page. "PTS/L 1024" indicates that the user has elected to have 1024 data points plotted per line. The user has the option of selecting from 1 to 2048 data points per line. 1024 plotted data points per line were selected to correspond to the 1024 points maximum size FFT that IDRS can calculate. "ISK 1" indicates that the user has selected to have no skip points and thus to have successive points plotted. If the user had selected ISK equal to 2, every other point would have been plotted. If ISK had been selected equal to 3, every third data point would have been plotted and so forth. A maximum ISK of 99 can be selected. "LPTS 0" indicates that the user selected to have zero overlap with each of the data lines 2 through 5 with respective to its previous line. The user may select to have an overlap of from 0 to the selected value of "PTS/L". "TIME 0:0:8:4576" indicates the starting time of the next line of data if it were displayed.

Computer-produced annotation also appears for each line of data in Figure 1-9. Using, for illustration, the annotation associated with and appearing above the first line of data, a starting time is given for that line of data: "TIME 0:0:7:1945". The format for the time annotation is,

Hours : Minutes : Seconds : Fraction of a Second (expressed in a number of sample interval counts).

It is important to point out that the indicated time refers to digitizing time and not signal time. Signal time is obtained by dividing ADC time by 16, the analog tape slow down factor. Next, "ESR = 20000" indicates that sample rate is known to be 20K samples per second. This was the sample rate of the analog-to-digital converter (ADC). Also, the maximum and minimum values are given for each line in units of ADC levels.

When more than 10 lines of data per page are requested, that "line" annotation is given for only the first line. The start time for the first line indicates the start time for the page.

The user accesses data by specifying the start time. The user advances from one page to the next by typing in "N" for next. The effective data sample rate is 320K samples/sec., and thus each line of data contains 3.2 msecs. of data referenced to real time. The parentheses around "quantities on display" annotation in figures in this section refer to real time as opposed to ADC time.

It is observed that the predetection digital signal shown in Figure 1-9 exhibits an envelope structure. This envelope structure is not symmetrical about the mean signal amplitude as would be the case for true amplitude modulation. The observed envelope is a beat relationship between the frequency of the digital signal and the sampling frequency. This occurs even though the sampling rate has been selected to meet the Nyquist sampling criterion as illustrated in Figure 1-10. The apparent amplitude modulation demonstrates an important factor. The point is that in order to obtain a plot of waveform samples that appear highly similar to the original analog waveform, the waveform must be sufficiently oversampled. Thus, oversampling of waveforms is desired for human analysis and interpretation. Even though the waveform plotted in Figure 2-9 is not sufficiently oversampled, an oversampling condition occurs as a result of subsequent digital signal processing.

Inspection of the waveform displayed in Figure 1-9 reveals a periodic structure with a period equal to approximately two lines of data or frequency of 156Hz. This periodicity is caused by a squelch side-tone FM modulation of the RT-524 signal, nominally at a 150Hz rate of deviation.

A display of the predetection waveform, as in Figure 1-9, serves as a coarse grain preview of the signal in time domain. A time domain preview, along with a frequency domain preview, allows the user to find a signal portion suitable for further processing. For example, if the user desires to develop a spectral plot requiring "n" lines of data to be averaged, he would want to make a coarse check to be certain that the signal was of sufficient duration to support an averaging of "n" lines of data. He may also wish to observe a coarse signal spectrum plot versus time.

1.4.2. Two-Dimensional Plot of a Power Spectral Density

Figure 1-11 shows an IDRS hardcopy plot of the power spectral density PSD of the signal displayed in Figure 1-9. The computer-generated annotation, "PS", at the top of the display indicates that the user has selected the power spectrum processing option. Furthermore, the user has asked for an average of 20 raw spectra, as indicated by "AVE:20". The annotation "LOG" indicates the selection of logarithmic power scale. Ten dB lines, referenced to the spectrum peak, are generated on the display to aid the user in his interpretation. It is noted that the computer-produced annotation indicates the frequency scale to be from 0 to 10KHz (the fold frequency). This frequency scale refers to ADC time. The signal frequency scale is obtained by multiplying the ADC frequency scale by a factor of 16. Thus, the signal frequency scale covers from 0 to 160KHz.

Each of the raw spectra making up the average PSD plot results from an FFT of a line of data containing 1024 time samples. Since a real signal is Fourier transformed, a power spectrum results that is symmetrical about the frequency origin. For this reason only, the positive half of the PSD is plotted, resulting in 512 frequency points per line. The resulting real time

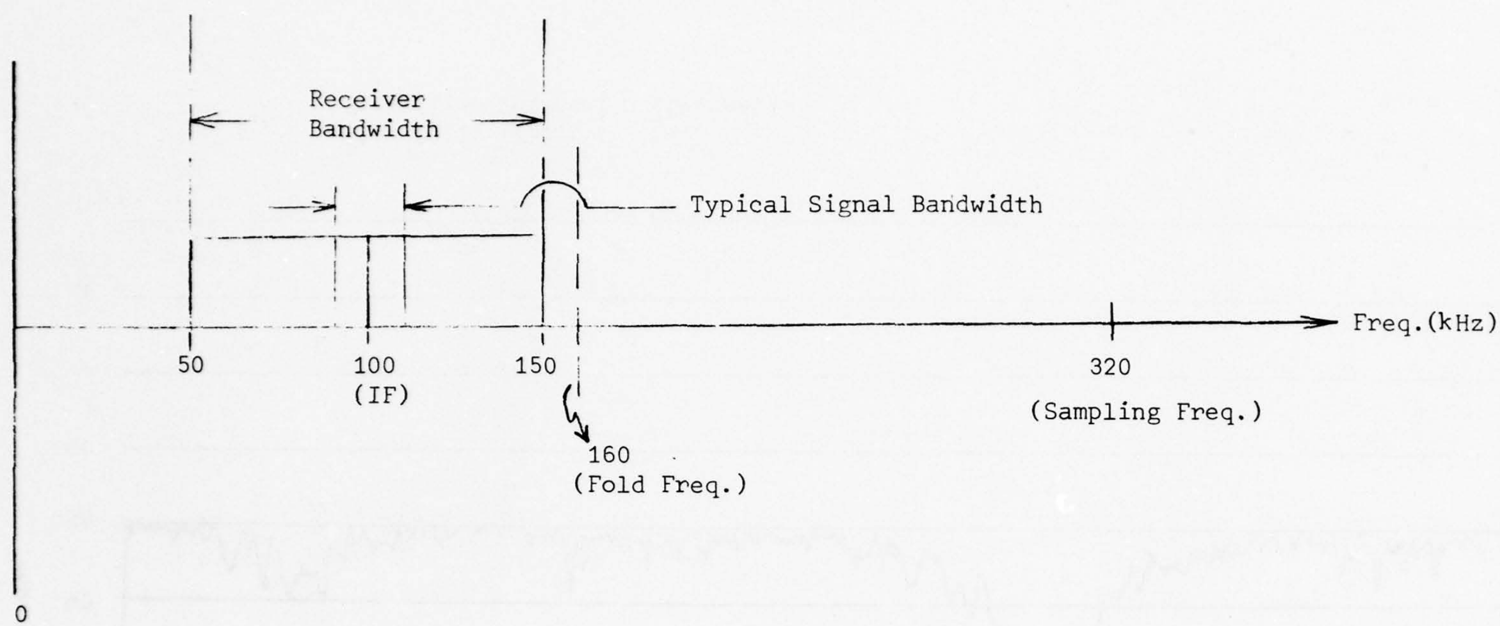


Figure 1-10 Relationship of Recorded IF Band and Sampling Rate

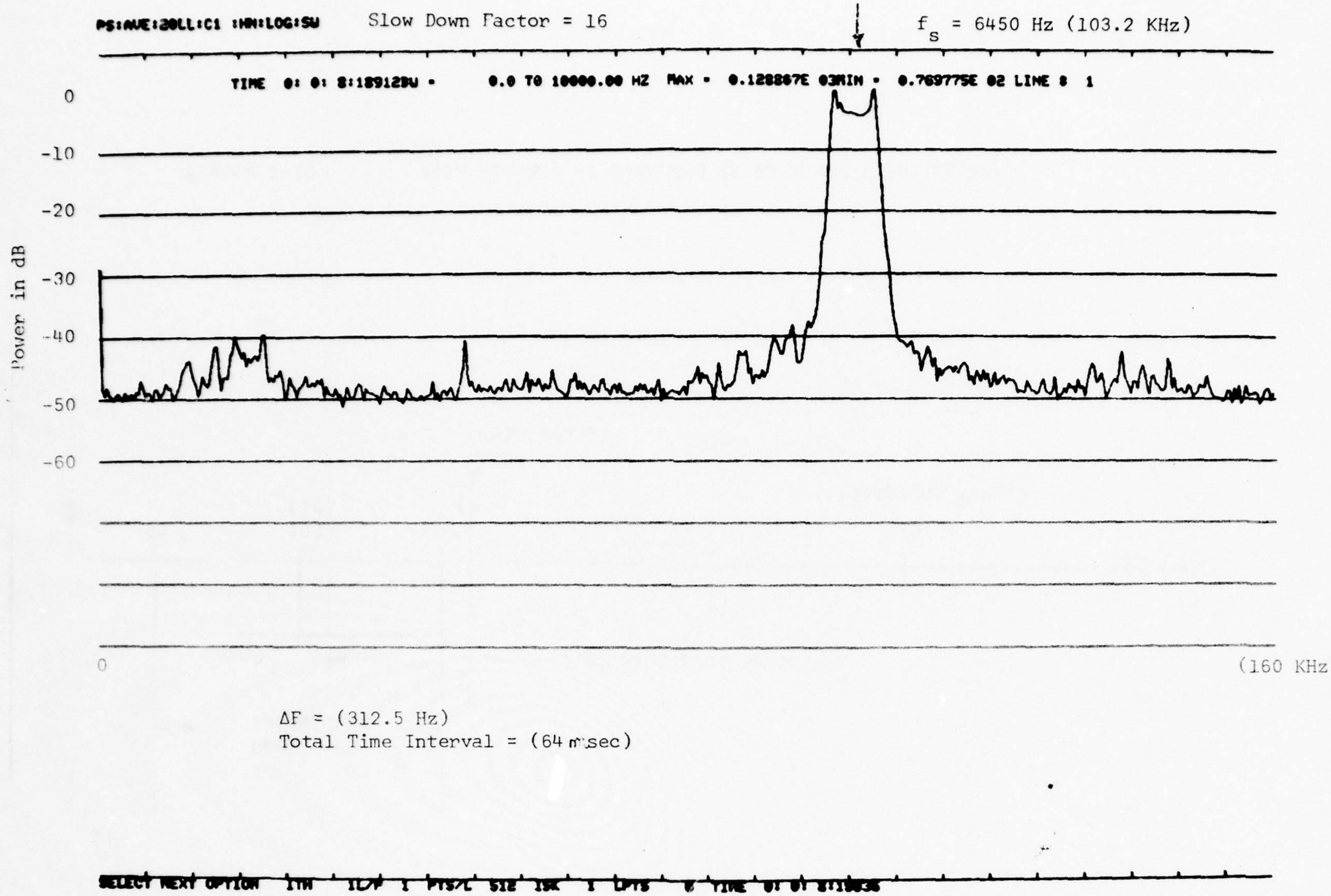


Figure 1-11 Display of Power Spectrum of a RT524 Predetection Signal With Hanning

frequency sampling interval, ΔF, is equal to the real time sampling fold frequency, 160KHz divided by 512, that is $\Delta F = 312.5$Hz.

The remainder of the computer-produced annotation has been defined previously.

The shape of the observed signal spectrum results from the RT-524 side-tone FM. Because this is wideband FM, the width of the spectrum is approximately equal to the peak-to-peak frequency deviation. This width is observed to be approximately 6KHz. The individual spectral lines, due to the approximately 150Hz sinewave modulation, are not resolved in the plot. The spectrum shape is, however, that of a classical wideband sinusoidal FM modulation. It is noted that sidebands other than due to the side-tone modulation appear to be 40 dB or more below the signal.

Since the actual carrier is not resolved, it has been estimated by the user to be the center of the signal spectrum. That is 6450 Hz (103.2 kHz in signal time). This number was then used as the shift frequency in the subsequent quadrature detection.

Also, the spectrum shown in Figure 1-11 was obtained from the waveform after a Hanning weighting had been accomplished. The annotation "HN" on the display indicates that the Hanning weighting had been applied. Figure 1-12 shows the spectrum of the same data without Hanning weighting. The advantage of the Hanning weighting is clearly evident in that additional detail of the spectrum is obtained at frequencies near the principal spectral lobe.

1.4.3. Plot of Power Spectral Density as a Function of Time

Figure 1-13 shows a display of the predetection power spectrum versus time on a linear power scale for the signal previously displayed at the same starting time. The user has selected 99 lines per page (L/P) and a line overlap (LPTS) of 900 samples. The selection of these parameters causes the display to have a three-dimensional appearance where relative power is observed as the height of the image and as a function of the frequency and time.

The parameters selected for the power versus frequency display shown in Figure 1-13 are such as to reveal nicely the squelch FM. The instantaneous frequency is observed to vary periodically. In general the power spectrum versus time plot is highly useful for displaying signal trends.

1.4.4. Plot of Signal Spectrum after Frequency Translation

Figure 1-14 shows the computer-generated PSD plot of the same signal sample but after frequency translation. Since a complex signal results from the frequency translation, both the negative and the positive halves of the calculated spectrum need to be displayed. Thus, in Figure 1-14, zero frequency appears in the center of the display and the shifted positive

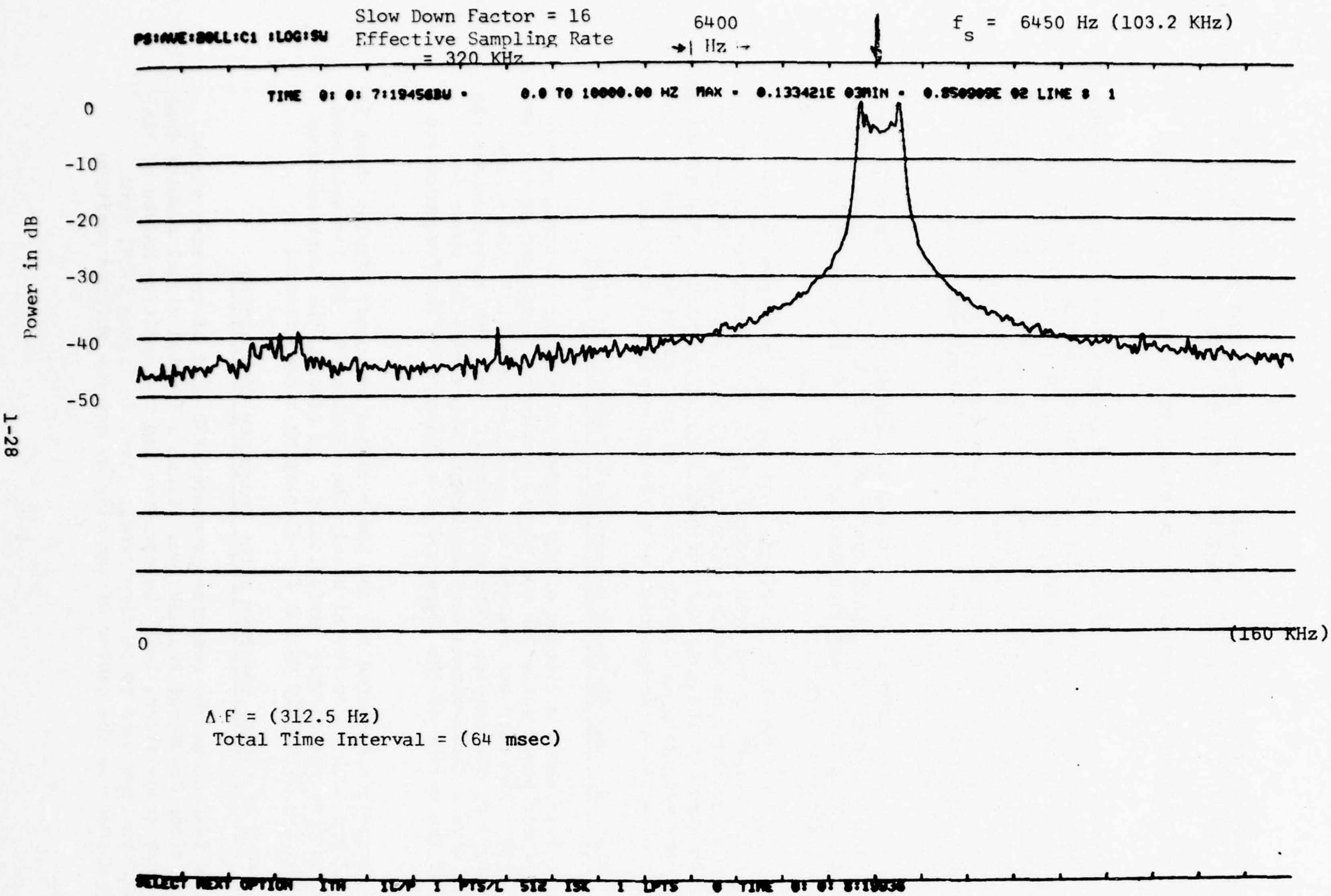


Figure 1-12 Display of Power Spectrum of a RT524 Predetection Signal Without Hanning

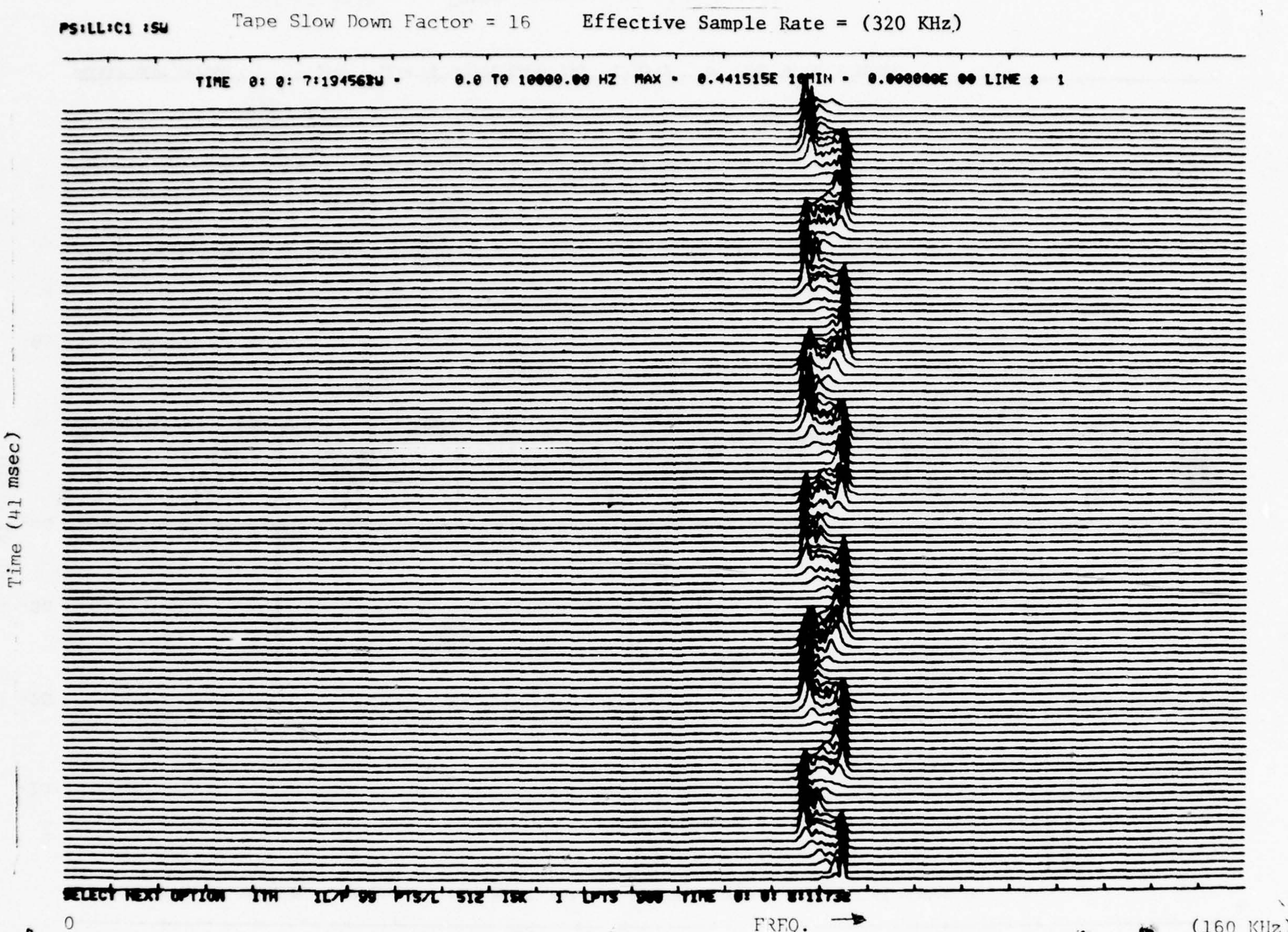


Figure 1-13 A Power Spectrum (One Linear Scale Per Line) vs. Time of a RT524 Predetection Signal

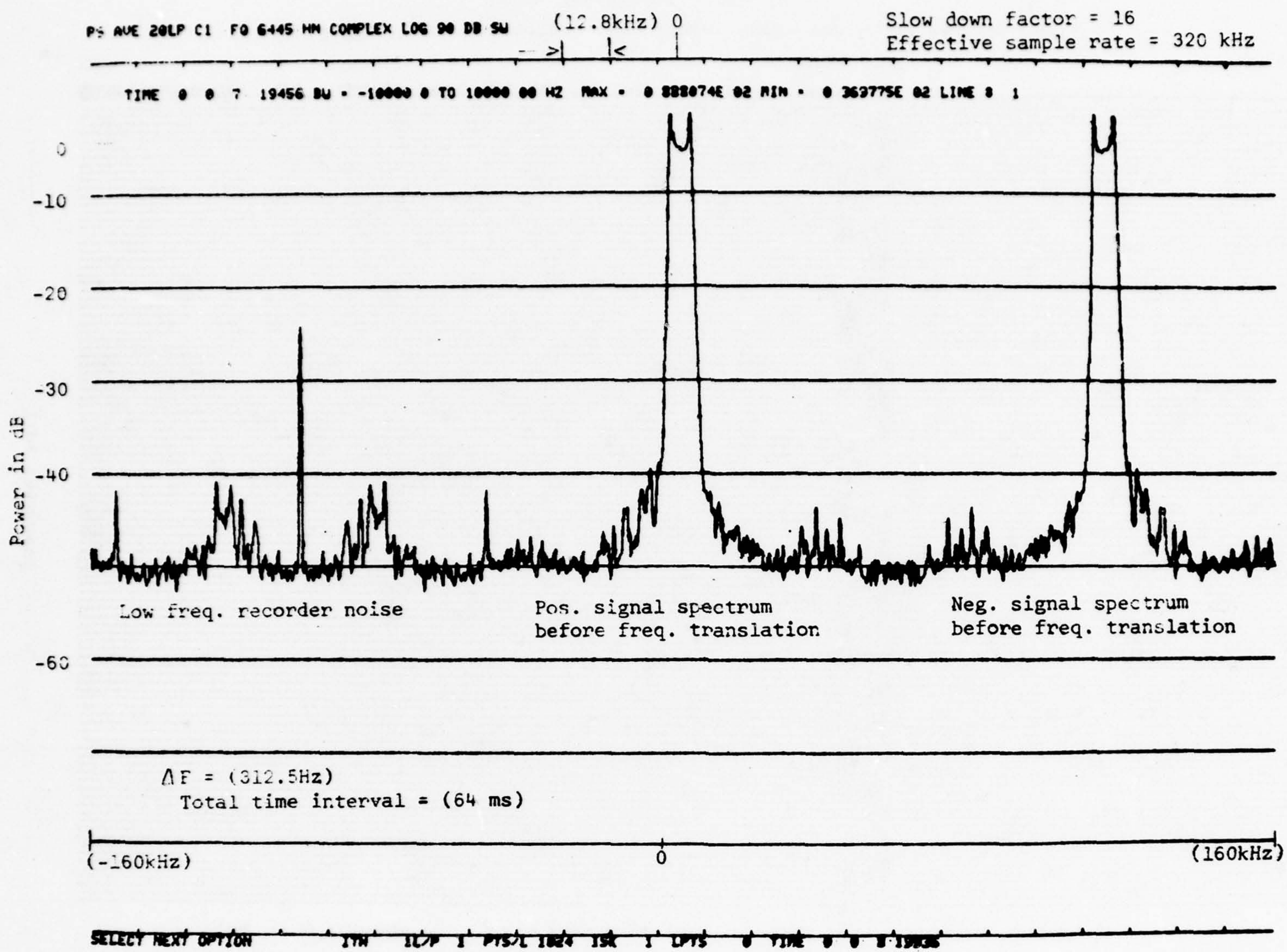


Figure 1-14 Power Spectrum of Signal After Frequency Shift

frequency portion is now centered at approximately zero frequency. The negative frequency portion of the signal spectrum is now shifted so that it appears in the positive frequency range below the positive fold frequency (160 KHz). The spectrum of the low frequency additive noise associated with the analog recorder is now shifted so as to appear in the negative frequency range. Except for the frequency shift and display of both the negative and positive frequency ranges, the PSD plot in Figure 1-14 was obtained with the same processing parameters as the PSD plot in Figure 1-11. It is noted that the frequency shift is annotated on the spectrum plot by "FQ 6445."

1.4.5. Waveform Plot of Frequency Translated Signal

Figure 1-15 shows a computer-generated waveform plot of the real part of the complex signal that results from the frequency translation. The five lines of data shown correspond precisely to the original sample shown in Figure 1-9 and the data starting time is the same as used in developing the above-discussed spectral plots.

Inspection of the waveform reveals high-frequency signal superimposed on a low-frequency signal. The high-frequency signal corresponds to the high-frequency spectrum, and the low-frequency waveform corresponds to the base-banded spectrum shown in Figure 1-14.

The effect of the squelch side-tone modulation is clearly evident in the low-frequency component of the waveform. Since the low-frequency spectrum is approximately centered at zero frequency, the FM side-tone modulation will cause the instantaneous frequency of the complex signal to alternate between positive and negative frequencies at the side-tone modulation rate. Thus, the signal goes through a time of "zero" frequency twice in each period of the modulation rate. These transitions through zero frequency are clearly evident in the real waveform plot, occurring once on each line of data.

The imaginary part of the complex waveform, if plotted, would appear quite similar to the real part.

1.4.6. Plot of Signal Spectrum after Frequency Translation and Filtering

The next step in the processing scheme was to filter the complex waveform that resulted from the frequency translation. The filtering accomplishes two things. First, the high frequency term is rejected so that the signal complex modulation envelope results in terms of in-phase [$m_I(t)$] and quadrature-phase [$m_Q(t)$] waveforms. Secondly, the bandwidth of the filter can be selected so that the filter serves as a predetection filter for rejecting unwanted noise and interference.

Figure 1-16 shows the computer-displayed characteristics of the designed filter. The annotation "FILTER TYPE=0" indicates that a Chebyshev filter was selected by the user; "PASS BAND=0" indicates that a low-pass filter form was

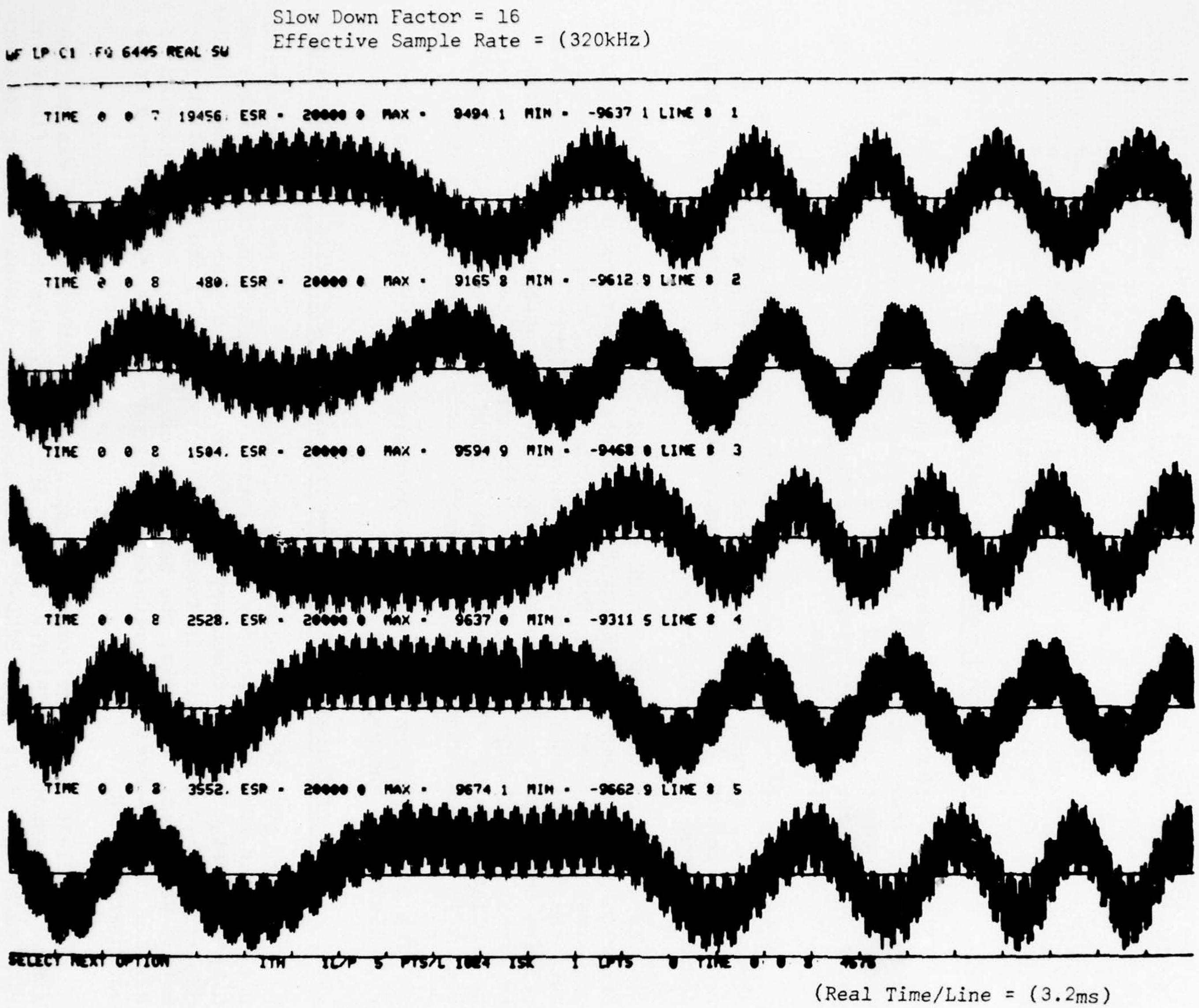


Figure 1-15 Waveform Plot of Real Part of Frequency Translated Waveform

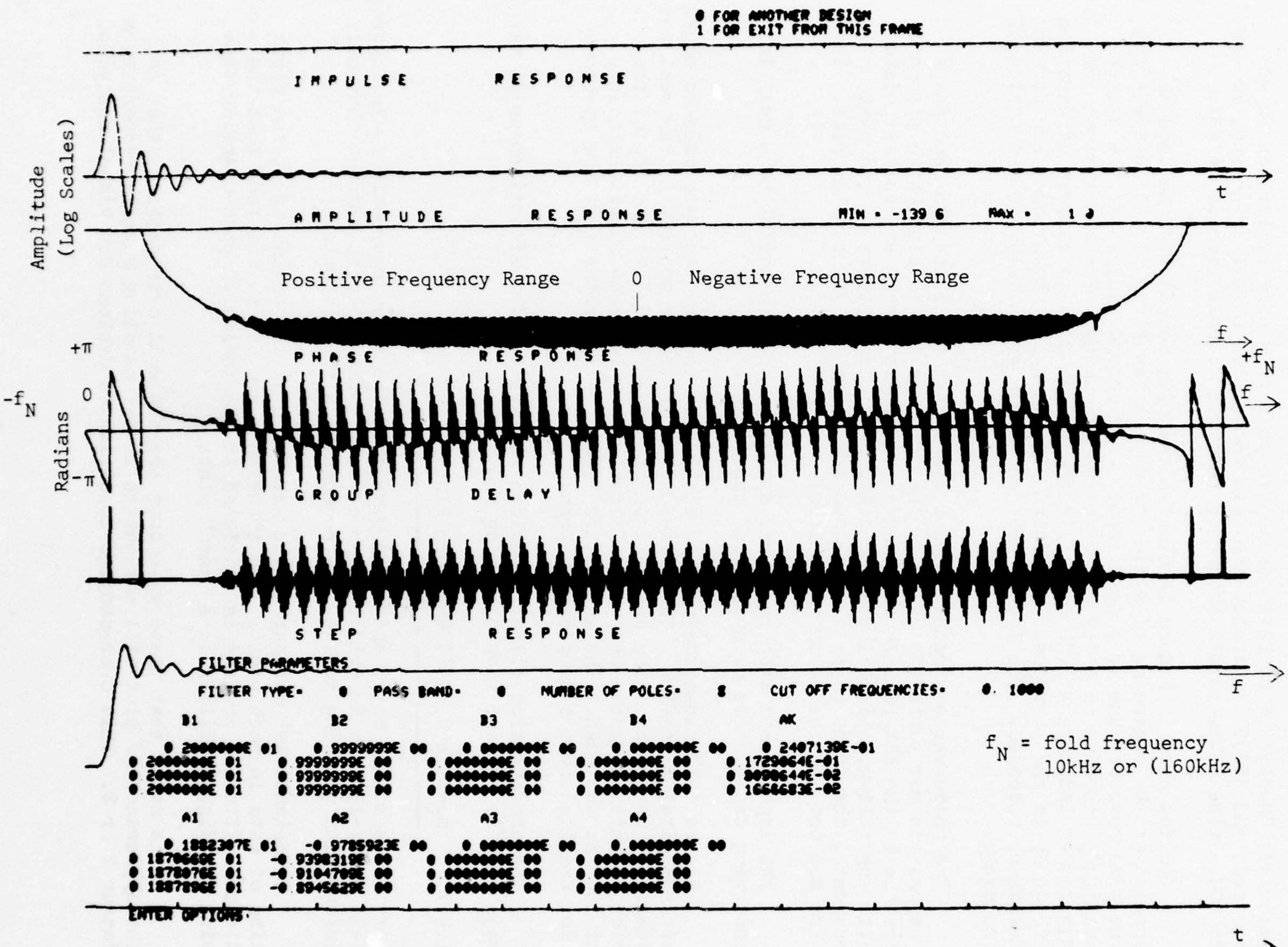


Figure 1-16 Display of Filter Characteristics

selected; "NUMBER OF POLES=8" indicates that an 8 pole filter design was elected; and "1000" indicates the selected filter cutoff frequency, referred to ADC - time. The cutoff frequency referred to real time is 16 kHz.

The top plot in Figure 1-16 shows the time response of an impulse applied at the beginning of a line of data. The fifth plot likewise shows the time response due to a step input. The second plot shows the filter amplitude response on a log scale versus frequency. The annotations "MAX=1.0 MIN=140.0" indicate a 1 db ripple in the passband and a maximum attenuation of 140 db in the stopband. The third plot likewise shows the phase versus frequency response. The fourth plot shows the group delay as a function of frequency.

The phase-versus-frequency plot shows that the filter in-band phase response is not perfectly linear. The folds in the in-band phase response are due to the RSX FORTRAN arc-tangent routine and thus are not associated with the filter. Since the group delay is the derivative of the phase response, the folds cause the in-band spikes in the group delay plot. These also are not indicative of the filter.

The impulse and step responses appear to be more useful to a user in observing signal time delay through the filter.

Figure 1-17 shows the plot of the PSD of the frequency-translated signal after digital filtering. The filter was identically applied to the real and imaginary waveforms. The annotation "FL2(CH LP 1000)" at the top of the plot, specifies that a Chebyshev filter with a low-pass cutoff frequency of 1000 Hz was used. The annotation "COMPLEX" indicates that the spectrum plot resulted from a complex waveform.

The predetection filter bandwidth was 32kHz. The effectiveness of the rejection is evident from observing the plot.

1.4.7. $m_I(t)$ and $m_Q(t)$ Waveform Plots

Plots of the $m_I(t)$ and $m_Q(t)$ waveforms are shown respectively in Figures 1-18 and 1-19. It is evident from comparing the plots to figures that only information relating to the low-frequency spectrum is retained.

Transients are evident in the waveforms at the beginning of the first line of data in both plots. The transients are due to the fact that the digital filtering process is initiated at the indicated page-starting time with the digital filter in a "zero" state.

It is noted that these waveforms are now well over-sampled, that is, the appearance of plot will not change if the sampling rate is reduced by a factor 2 or 3. This over-sampled condition is desired for visual analysis.

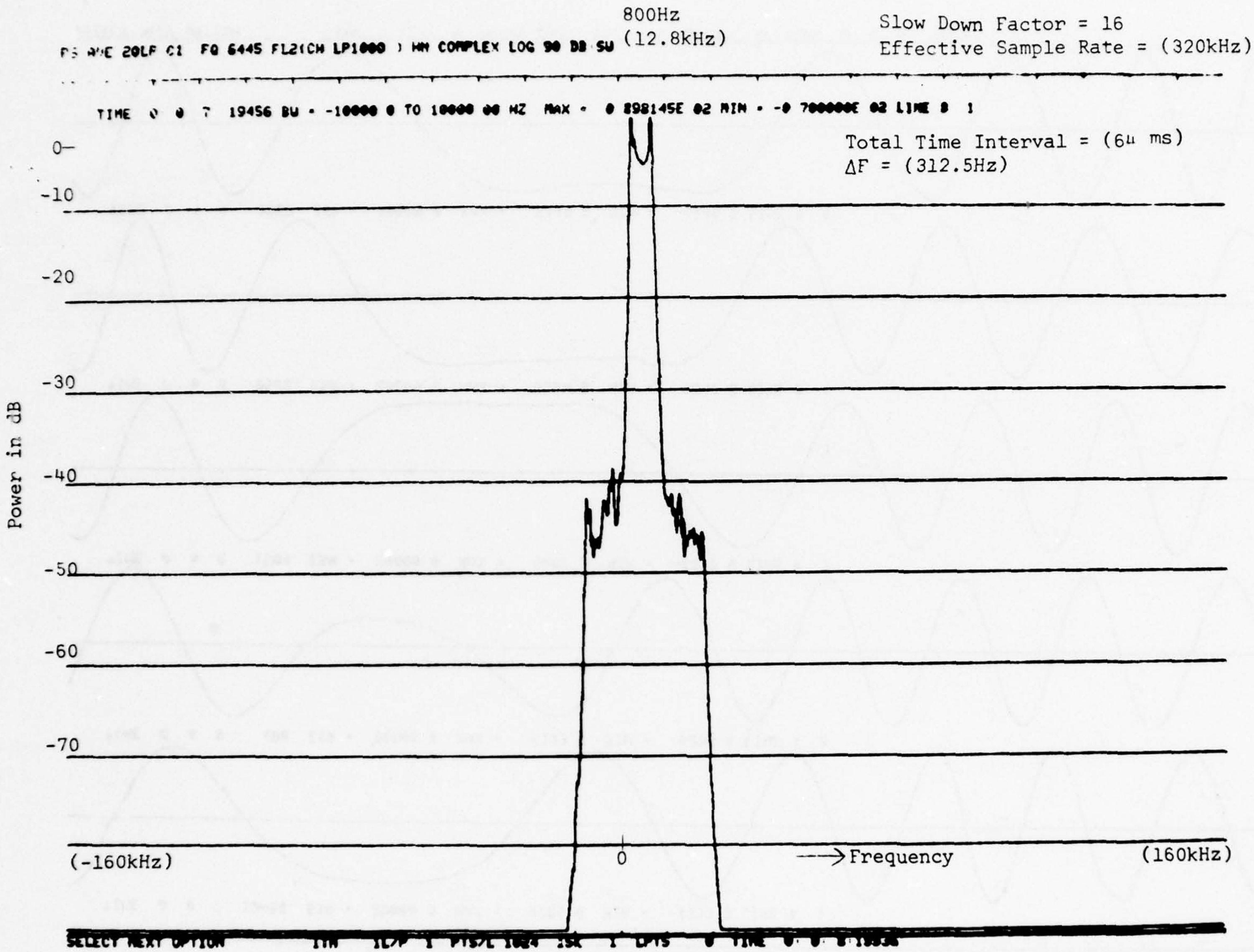


Figure 1-17 Extracted Power Spectrum of Complex Modulation Envelope
(Filter Parameters: 8 Pole Lowpass Chebyshev with Cutoff Equal to 1000Hz)

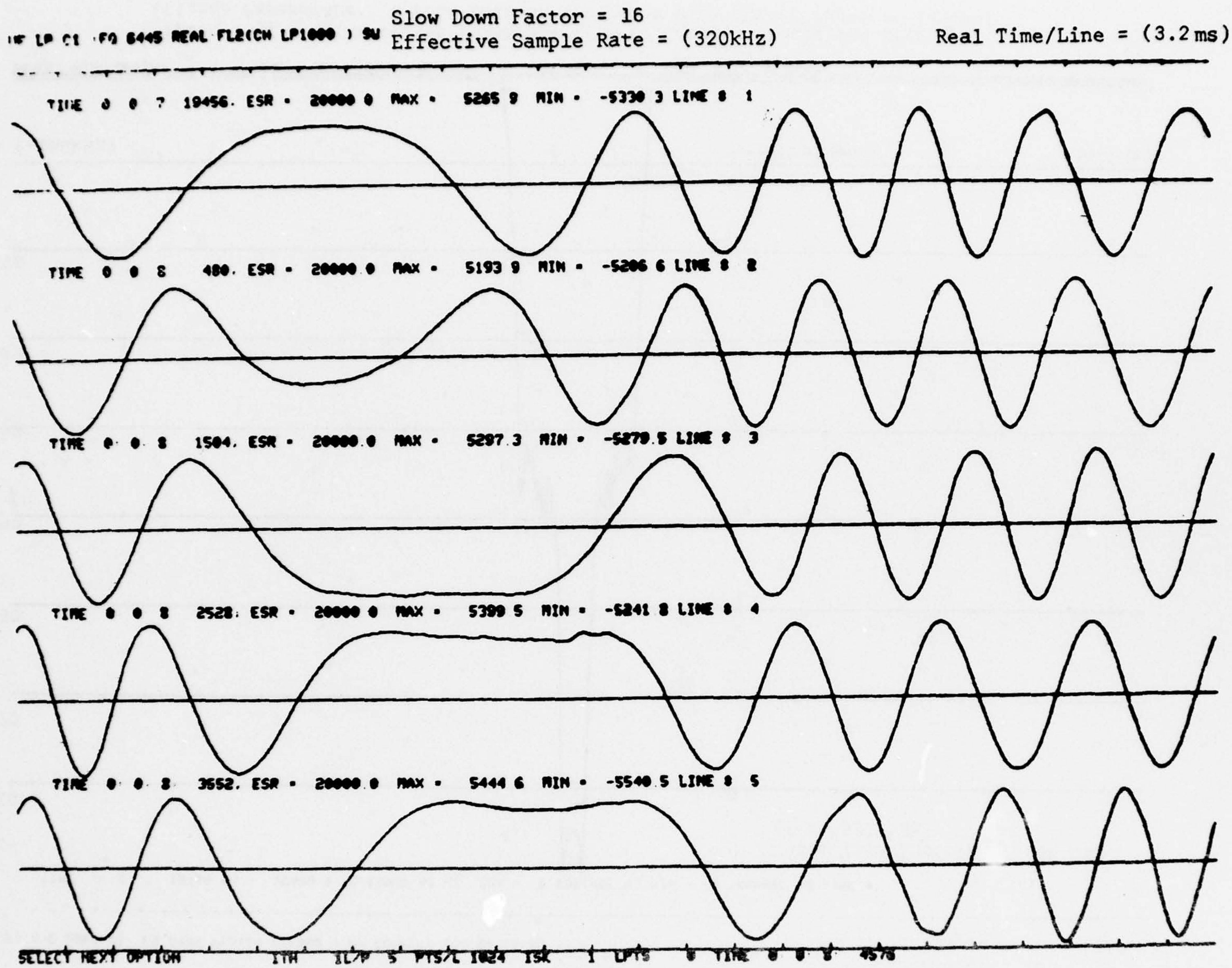


Figure 1-18 In-Phase Waveform Plot

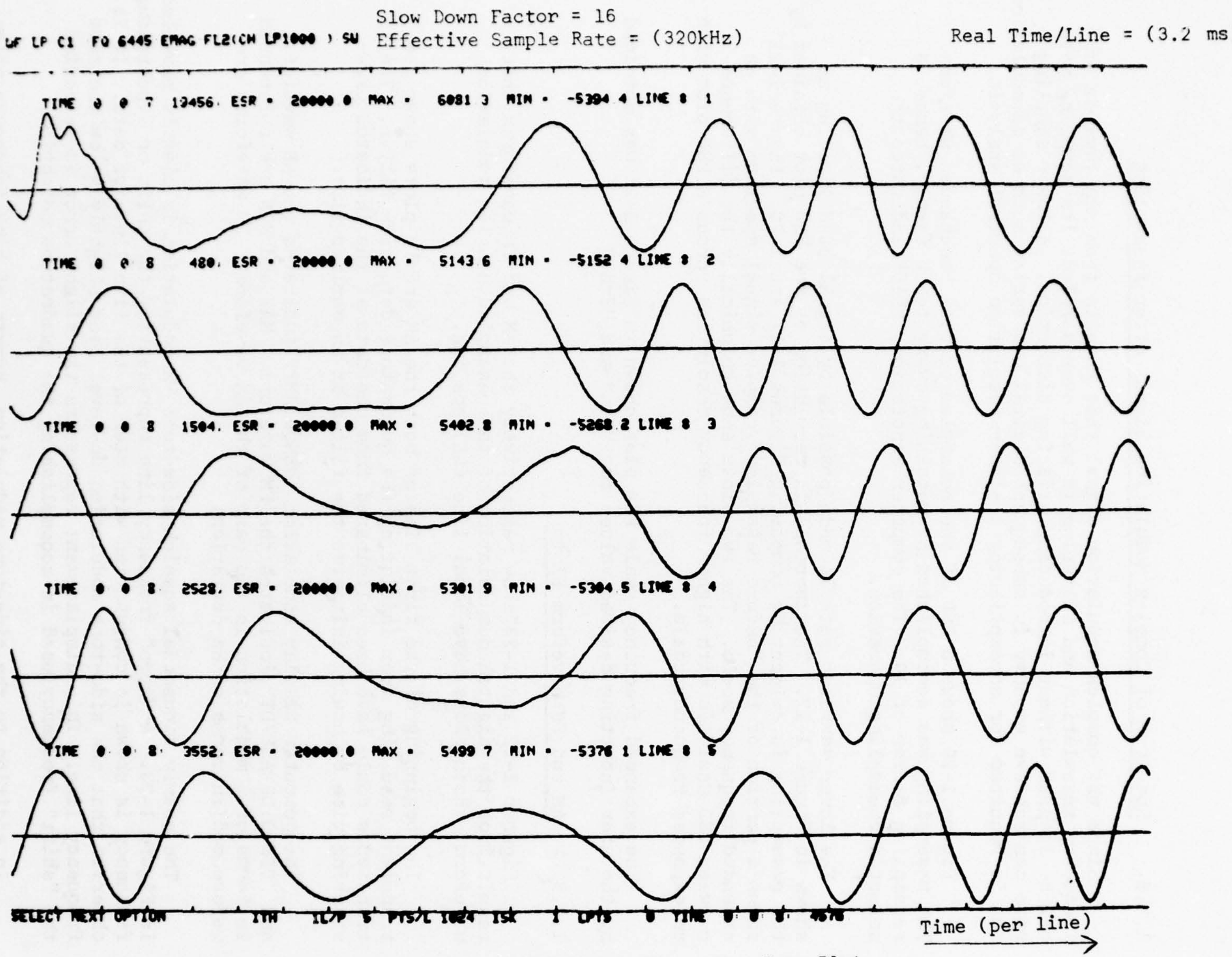


Figure 1-19 Quadrature Phase Waveform Plot

1.4.8. Spectrum of Complex Modulation Signal after Resampling

Since the complex modulation signal that results from the process of frequency translation and filtering is well over-sampled, its sampling rate can be lowered without introducing aliasing distortion. A lower sampling rate can achieve economy in subsequent digital processing such as demodulation and is required for accomplishing a higher resolution spectral analysis.

Figure 1-20 shows a PSD of the resampled complex basebanded waveform. The resampling was accomplished by retaining every tenth sample, hence, a resampling factor of 10. The computer annotation "RS10" indicates the selected resampling operation.

The lower sampling rate is made possible by signal band limiting as shown in Figure 1-17. The increase in resolution of the PSD plot obtained by the resampling is evident by comparing Figures 1-17 and 1-20. Figure 1-21 shows a portion of the "before resampled" complex signal spectrum with an expanded frequency scale. The two plots show graphically the difference between PSD analysis with higher frequency resolution versus a PSD plot with an expanded frequency scale.

The expanded frequency scale PSD plot shown in Figure 1-21 was obtained by the user inputting desired values for "Ith" and "Jth."

1.4.9. AM and FM Waveform Plots

Figures 1-22 and 1-23 show respectively the AM and FM waveforms that result from the digital demodulation of the resampled complex modulation waveform. Both plots have local line (LL) scaling.

The beginning of the first line of both the AM and FM plots shows the transient resulting from initiating the quadrature detection filter. The transients could have been eliminated from the data at the indicated page-starting time by having initiated the filter at an earlier time.

The computer display annotation "DEMOD:AMP" indicates the AM waveform, and "DEMOD:DPHASE/DT" indicates the FM waveform. MAX and MIN are referenced in terms of % modulation in the case of the AM waveform. FM waveforms are referenced in terms of Hz deviation.

The nearly sinusoidal squelch side-tone FM modulation is clearly revealed in Figure 1-23. A "zero" frequency line representing the shift or synchronous frequency is shown in conjunction with each of the five lines of data. It is observed that the side-tone modulation dc-level lies slightly below the zero frequency line. This displacement represents the slight error in selecting the "shift" frequency used in accomplishing the quadrature detection.

In addition to the side-tone modulation, bursts of high frequency noise are also observed in the FM waveform. Calibrations have shown that these

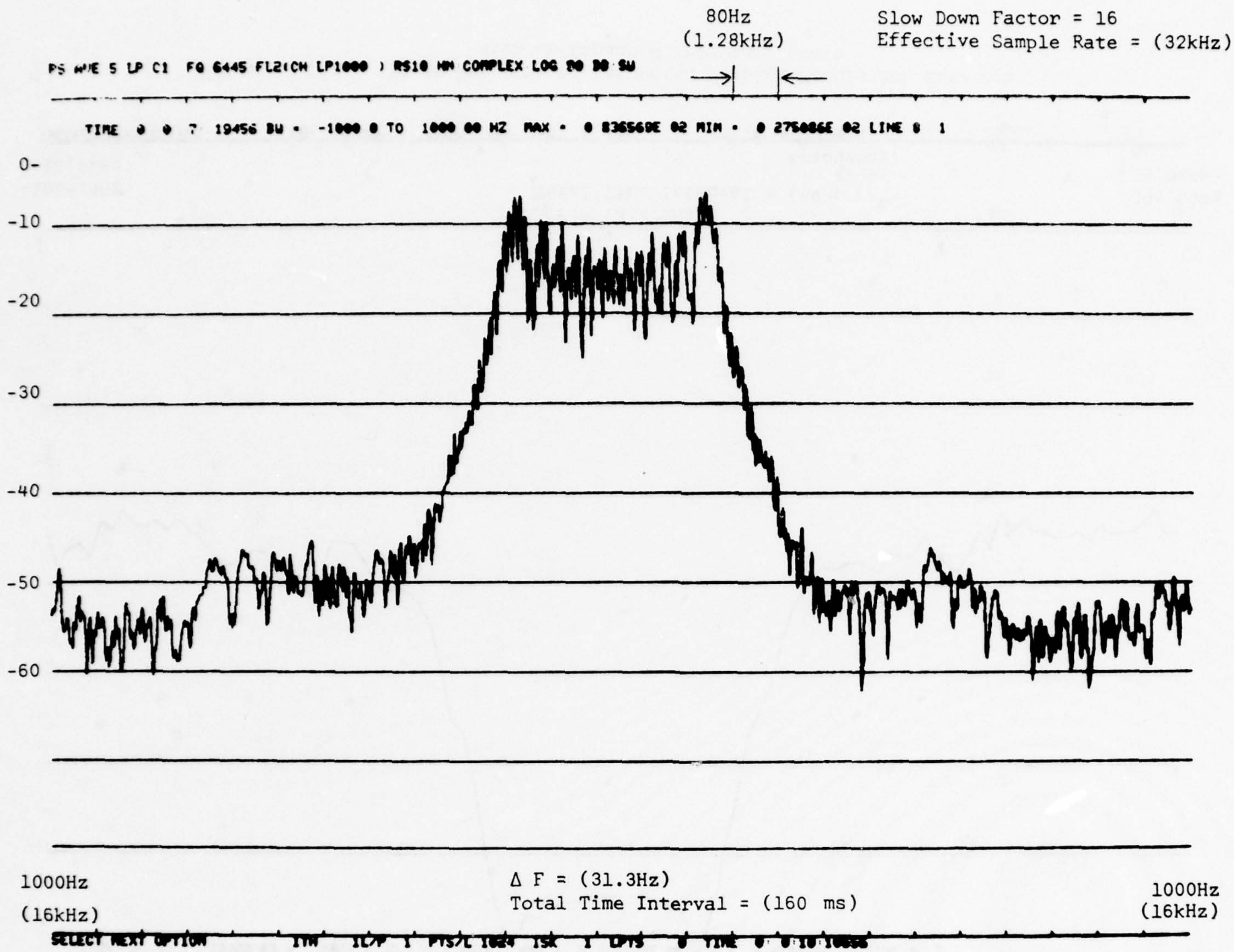


Figure 1-20 Power Spectrum of Extracted Complex Modulation Envelope After Resampling by a Factor of 10

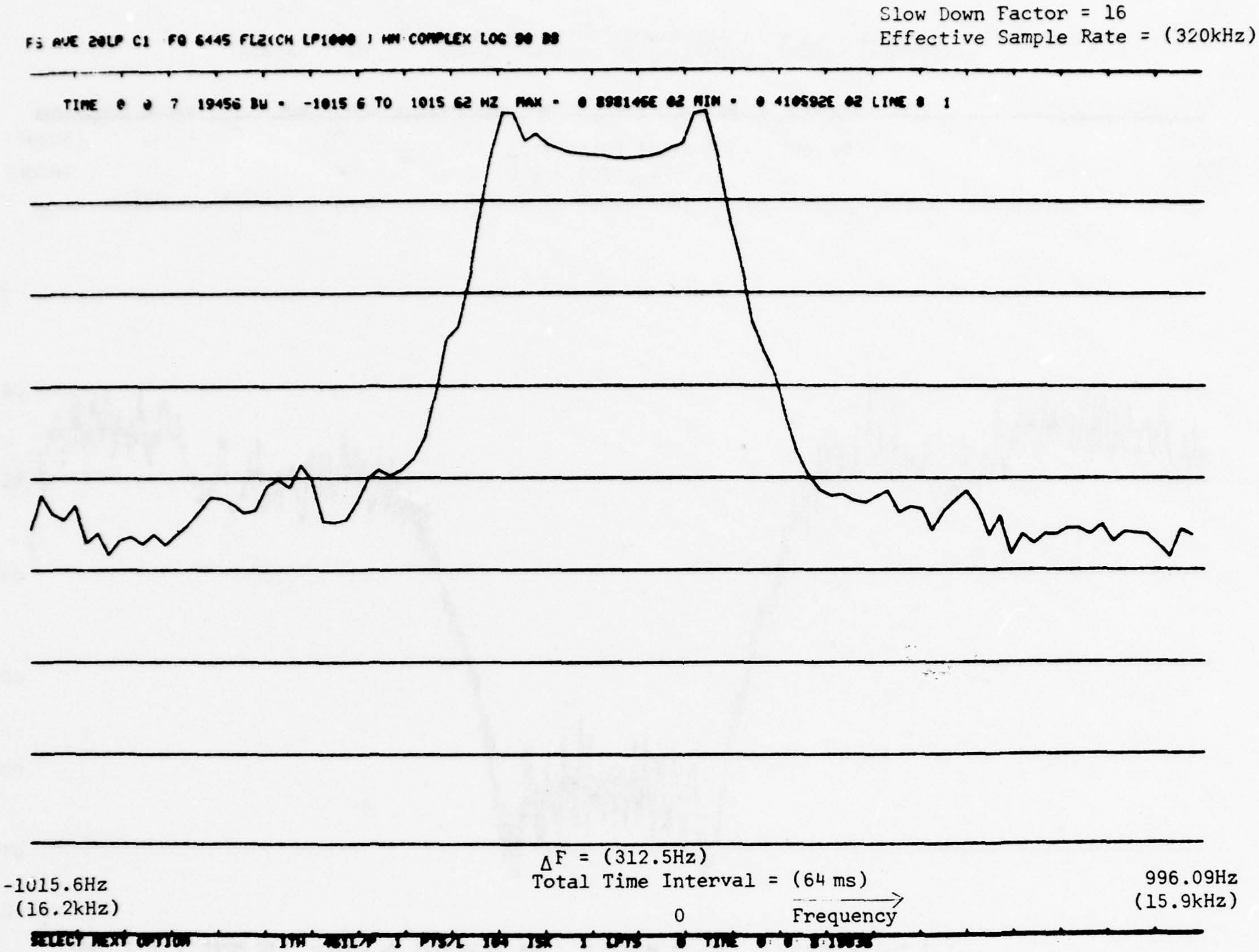


Figure 1-21 Power Spectrum of Extracted Complex Modulation Envelope
With an Expanded Frequency Scale

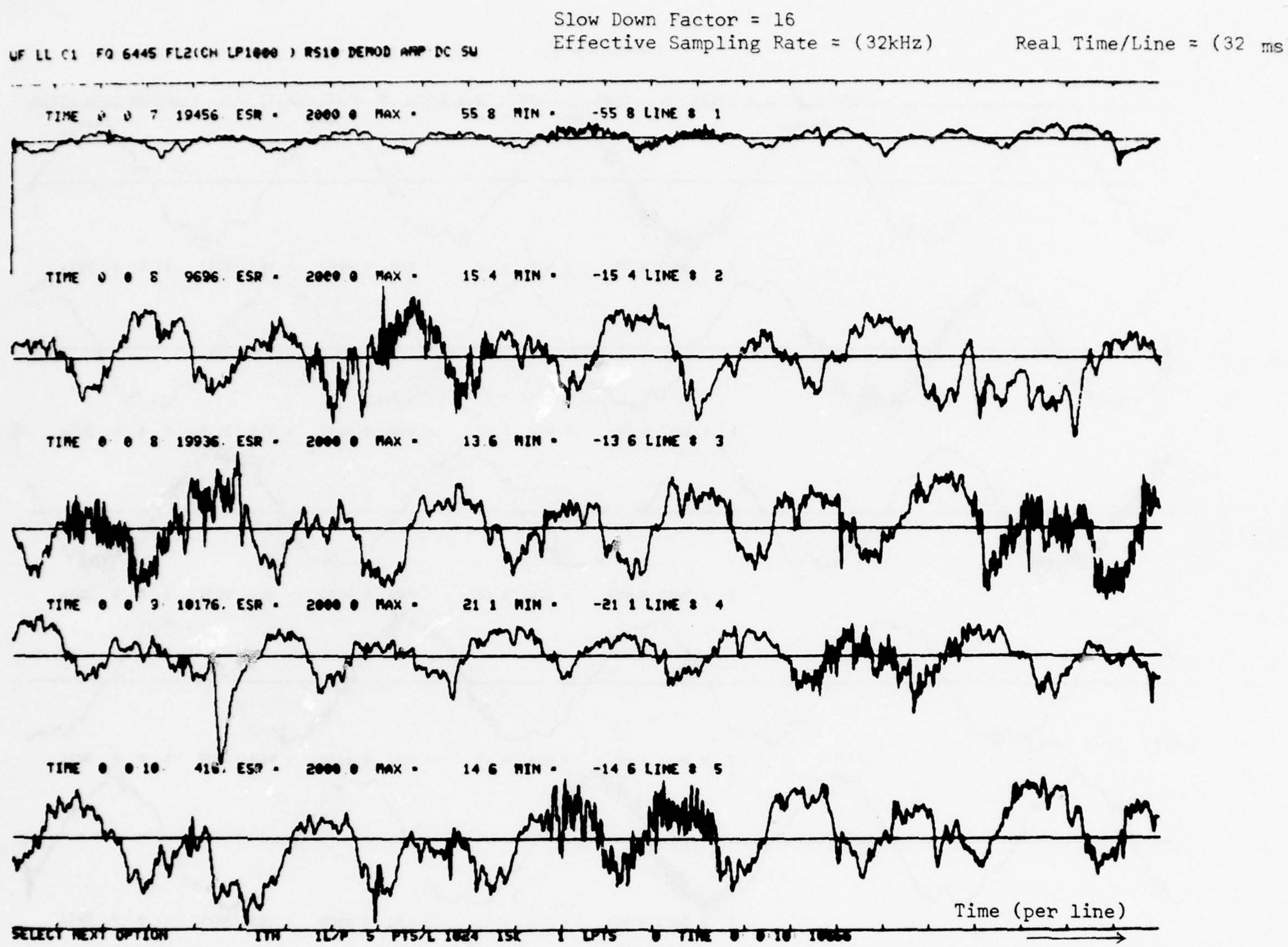


Figure 1-22 Extracted AM

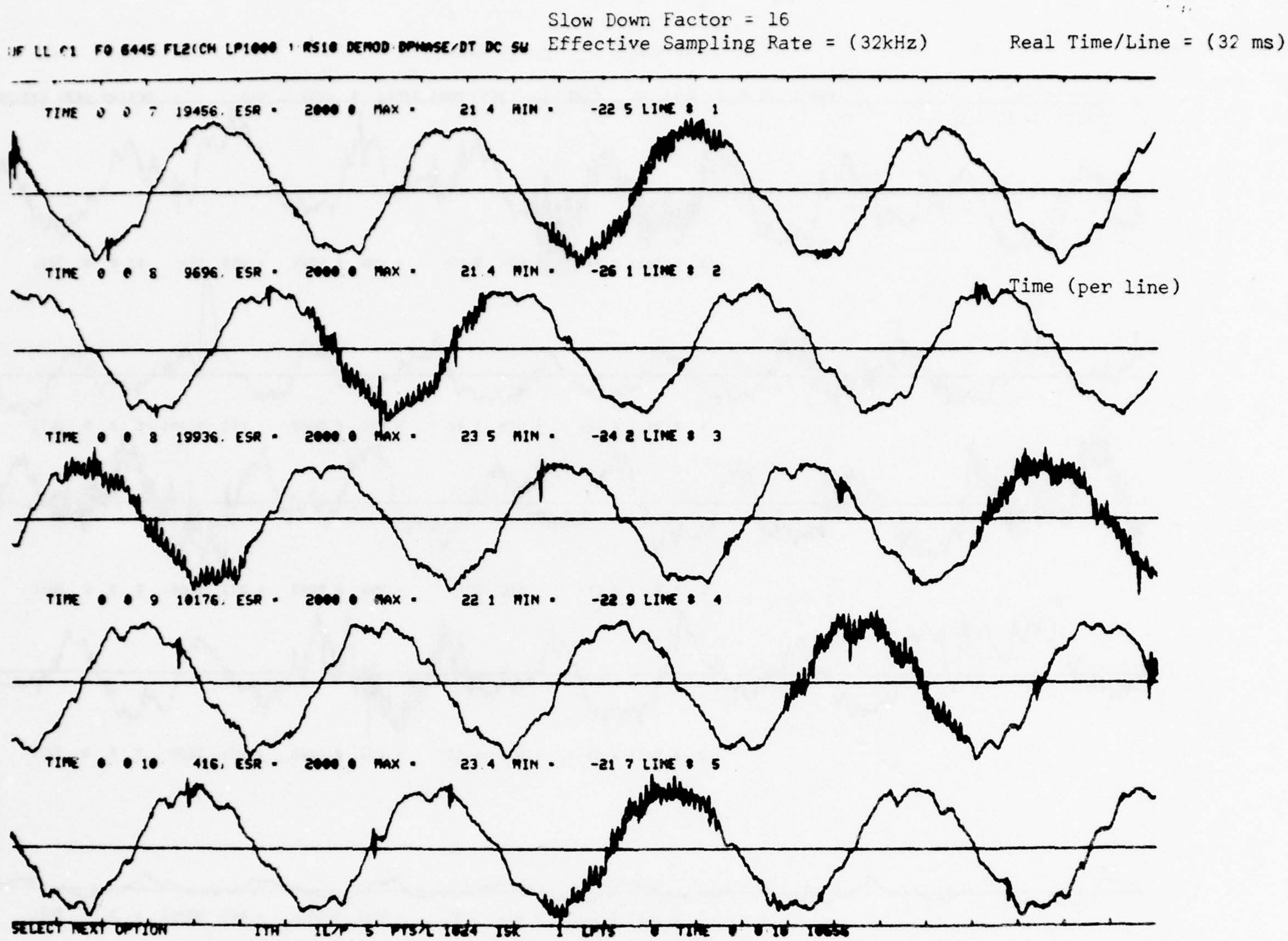


Figure 1-23 FM Waveform

noise bursts were introduced by a malfunctioning analog-to-digital converter (ADC). Currently, digitizing is being done at PAR's ADC facility.

The squelch side-tone modulation parameters can be estimated from the FM waveform plot. The rate of FM is calculated first. There are 24 cycles of the modulation shown and these occur during an interval of 0.16 seconds (signal time). Hence, the rate of modulation,

$$\text{rate} = \frac{24 \text{ cycles}}{0.16 \text{ sec}} = 150 \text{ Hz.}$$

The peak-to-peak frequency deviation is calculated next. The waveform "MIN" and "MAX" values indicated on Figure 1-23 are in Hz deviation referenced to digitizing time. To compute the real-time deviation we multiply by 16 or

$$f_{pp} = 16(\text{MAX-MIN})$$

where f_{pp} is the peak-to-peak frequency deviation.

It is noted that the factor 16 in the above calculation accounts for the scaling required because of 16 times tape play back slow-down. Hence, the peak-to-peak frequency deviation is equal [2924 - (-3911)]Hz or 6835 Hz.

Both the FM rate and peak-to-peak frequency deviation measurement values are reasonable for the RT-524 squelch modulation.

The AM waveform plotted in Figure 1-22, also reveals a periodic waveform, although embedded with much more noise than in the FM waveform. Inspection of the periodic AM component shows that its modulation rate is twice that of the FM side-tone modulation. This AM is a cross-modulation from FM that occurs as an unintentional property of the RT-524 transceiver. It is noted that the bursts of high frequency noise observed in the FM waveform are also evident in the AM waveform. The annotation "MAX" and "MIN" show that the average % AM modulation over one line is approximately 15%.

Finally, it is noted that both the FM and AM waveform plots were obtained from resampled waveforms, where the resulting effective sample rate was 32K samples/sec. The waveforms with this sample rate appear sufficiently over-sampled for analysis.

1.4.10. FM and AM PSD Plots

Figures 1-24 and 1-25 show respectively plots of FM and AM power spectral density (PSD) on a dB scale. Max and Min are referenced in dB relative to 1% modulation for the AM spectrum. The FM spectrum is referenced in dB relative to 1Hz deviation.

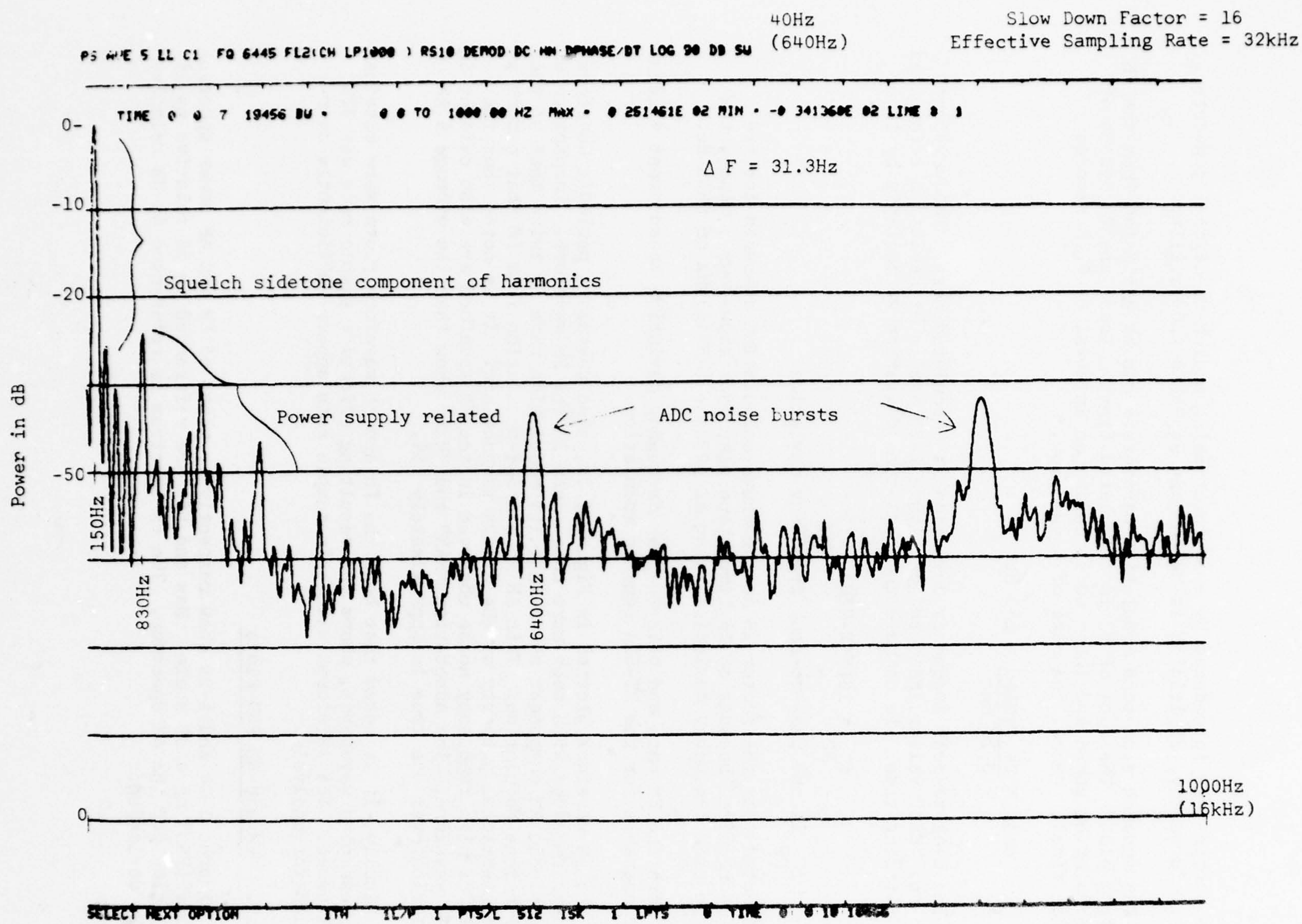


Figure 1-24 Power Spectrum of Extracted FM Waveform

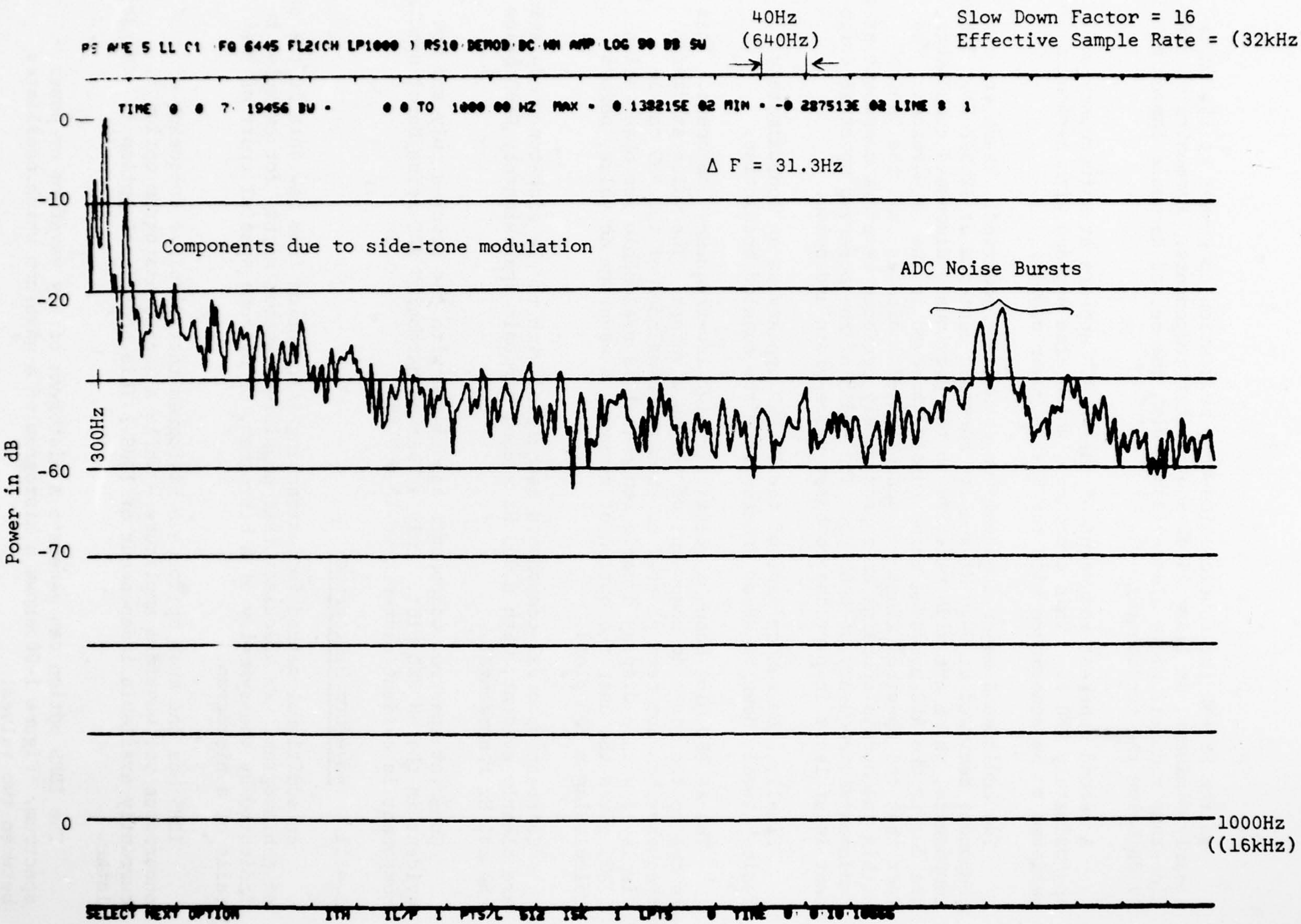


Figure 1-25 Power Spectrum of Extracted AM Waveform

In the FM PSD the squelch side-tone modulation component at 150 Hz is clearly dominant, at least 25 dB above other components. Harmonics of the side-tone modulation are also evident, with the second harmonic approximately 27 dB below the fundamental.

A second coherent component of the FM PSD appears at a frequency of approximately 830 Hz. This component, which also appears with harmonics, is believed to be associated with the RT-524 power supply.

Two additional major spectral components are observed. These are a component centered at 6400 Hz and its harmonic centered at 12,800 Hz. These components, which are wide relative to the previously discussed components, are due to the ADC produced noise bursts observed in the FM waveform. The fact that the spectral lobes are concentrated indicates that the "noise" within the bursts contains an approximately periodic waveform component at a fundamental frequency of 6400 Hz. The width of the observed spectral component is at least in part due to burst pulse-width and noise.

Finally, the random part of the FM PSD appears as an approximately "white" level, down 50 dB or greater from the squelch modulation.

The AM PSD plot shows a relatively high low-frequency component. This is due to the large dc component of the AM waveform. The value at "zero" frequency has not been plotted so that the remainder of the PSD can be displayed with a display dynamic range that is reasonable for observation (IDRS gives the user the option of removing a waveform dc value, before calculating a PSD plot).

Coherent spectral components associated with the FM side-tone modulation are clearly evident, with a 300 Hz second harmonic approximately 8 dB below the 150 Hz fundamental.

Coherent spectral components associated with the power supply are not evident in the AM PSD plot. Only a "second harmonic" ADC noise burst spectral component is evident centered at 12,800 Hz.

1.4.11. Waveform Histograms

An additional method for extracting information from raw data is the use of a histogram. We speculate that possible features might, for example, be indicated by the symmetry of a histogram, or by some special trait of the tails of a histogram.

IDRS has one such option - a histogram that displays frequency of occurrence vs. waveform amplitude - while a second histogram option is currently available independent of IDRS. This additional option is discussed later.

The IDRS option can generate a histogram of any waveform or power spectrum. Figure 1-26 shows a histogram of a waveform which oscillates between two values.

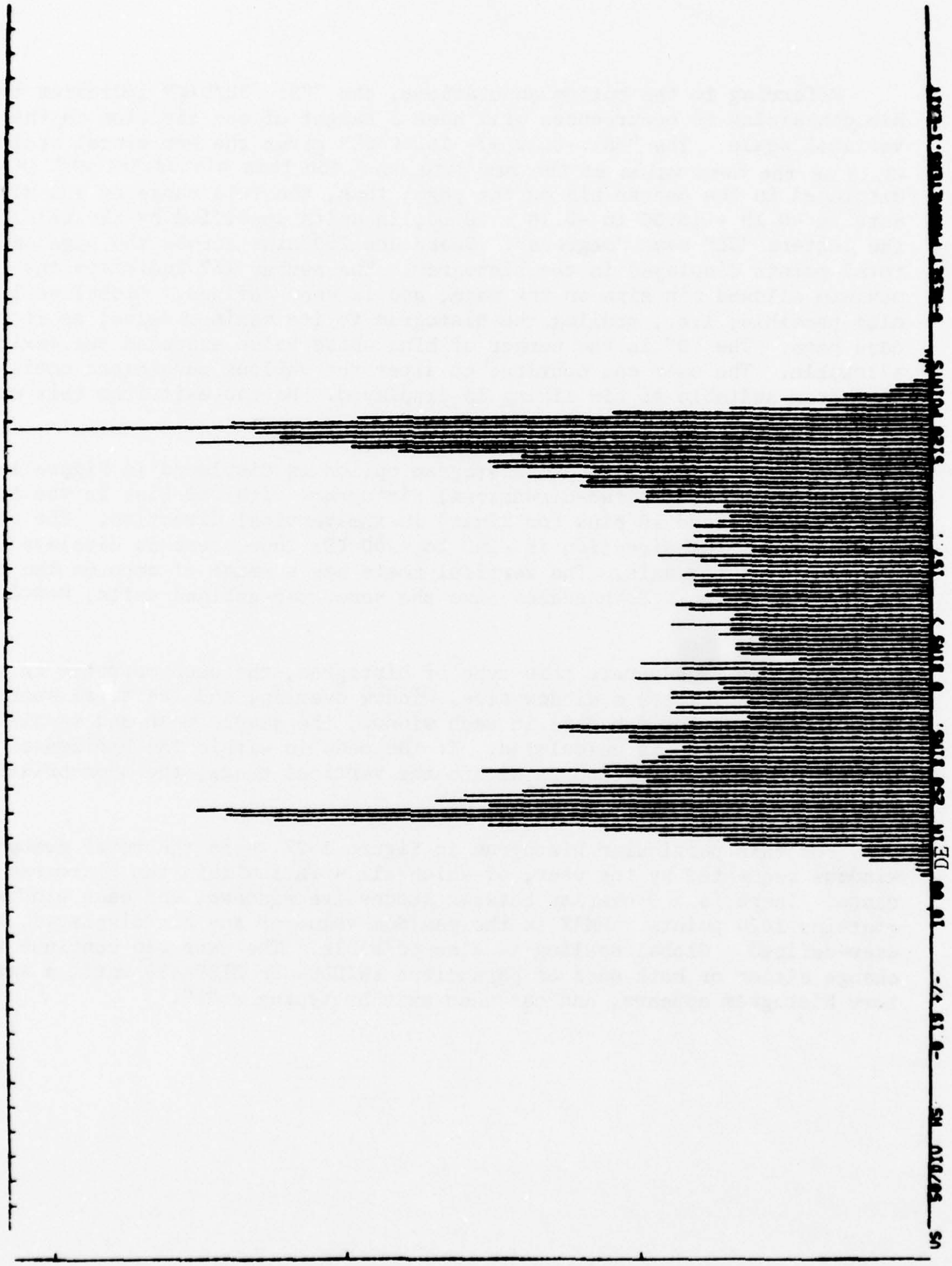


Figure 1-26 Histogram of FM Waveform

Referring to the bottom annotations, the "VS: 50/DIV" indicates that a bin containing 50 occurrences will have a height of one division on the vertical scale. The "HS: -0.19 +/- 10.00 DE" gives the horizontal scale: -0.19 as the mean value of the raw data used for this histogram, and is displayed in the center bin on the page; thus, the full range of the displayed data is -0.19 - 10.00 to -0.19 + 10.00, in units specified by the user. Here the letters "DE" mean "degrees". There are 250 bins across the page and 5120 total points displayed in the histogram. The number 157 indicates the maximum allowed bin size on the page, and is user-defined. Global scaling is also possible; i.e., scaling the histogram to its maximum value, as is the case here. The "0" is the number of bins whose value exceeded the maximum allowable. The user can continue to alter the various parameters until a histogram suitable to his liking is displayed. He can exit from this option by typing a "1".

An example of the second histogram option is displayed in Figure 1-27. This is effectively a two-dimensional histogram, with 250 bins in the horizontal direction, and 10 bins (or lines) in the vertical direction. The range in the horizontal direction is -200 to +200 DE; thus, zero is displayed in the center of the axis. The vertical scale has a range of zero on the bottom to 100 at the top. Both scales have the same user-defined units, namely degrees.

In order to generate this type of histogram, the user operates on the raw data. He selects a window size, window overlap, and the total number of windows. Then, for the data in each window, the sample mean and sample standard deviation is calculated. If the mean is within the horizontal range and the standard deviation is within the vertical range, the appropriate bin is incremented by one.

For this particular histogram in Figure 1-27, 4 is the total number of windows requested by the user, of which all 4 fall within the histogram range. There is a 0 overlap between successive windows, and each window contains 1024 points. HMAX is the maximum value of any bin displayed, and is user-defined. Global scaling is also possible. The user can continue to change either or both sets of parameters (WINDOW or DISPLAY) until a satisfactory histogram appears, and can then exit by typing a "1".

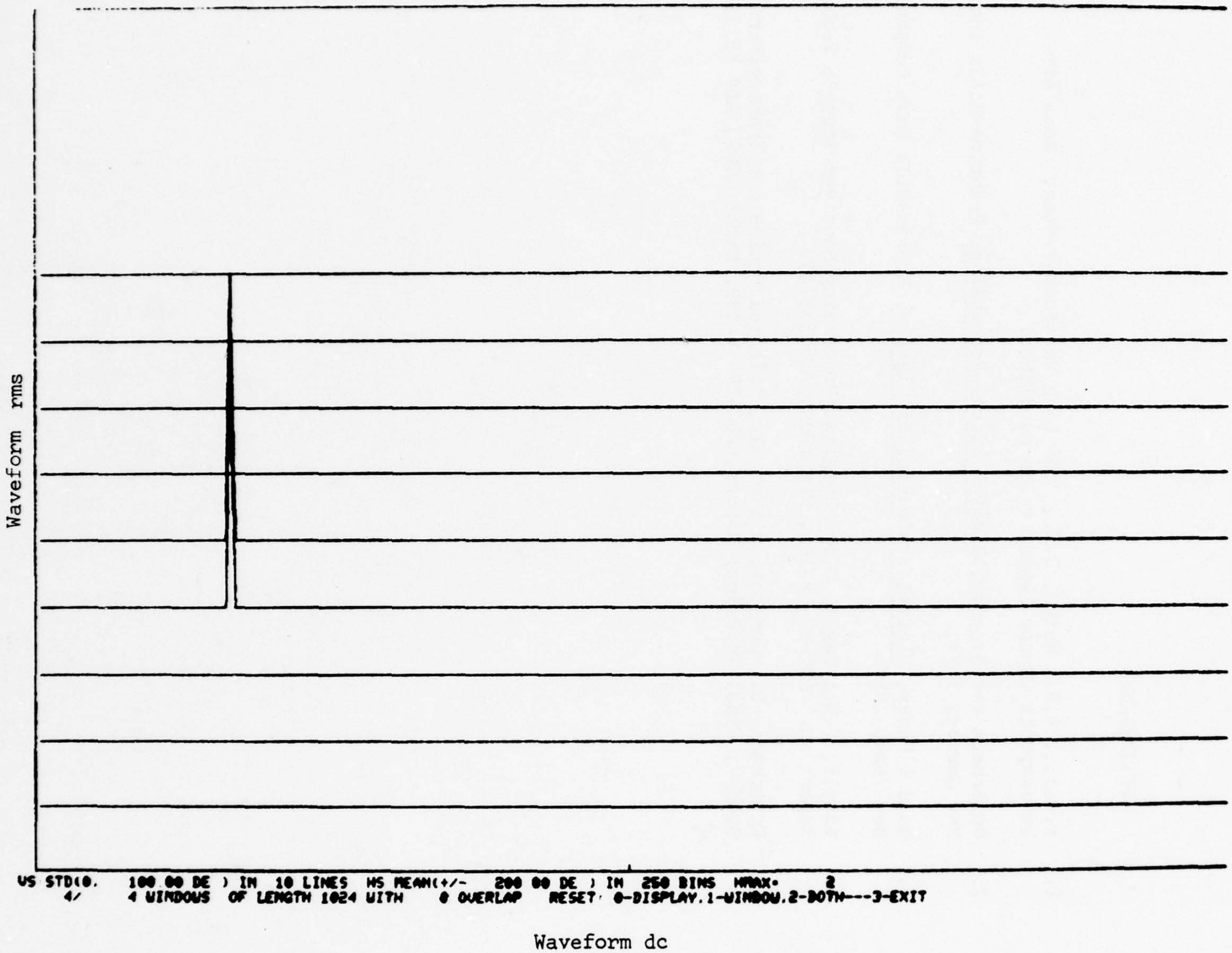


Figure 1-27 Histogram of Waveform rms vs. dc Level

1.5. REFERENCES

1. Proctor, A.H.; White, D.D., "The Long Waveform System," Rome Air Development Center Report to be published.

2. Oppenheim and Schafer, Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

3. Gold & Rader, Digital Processing of Signals, McGraw-Hill Book Company, New York, 1966.

4. Natali, F.D.; Magil, D.T., "Digital Processing Receiver Study", Technical Report No. RADC-TR-68-163, May 1968, AD# 835767.

5. Fritchman, B.; Gumacos, C.; et al., "Digital Equivalent Transceivers Study", Final Technical Report RADC-TR-68-539, March 1969, AD# 851364.

SECTION 2

IDRS OVERLAY STRUCTURE

Due to the limited amount of core space (32k) available for any one program, the IDRS system has been segmented into a number of subroutines and overlays. A portion of the available core is reserved for the Executive program, subroutines which are needed by all overlays, and a common area in which variables common to all overlays reside. The remaining core is available for overlay use.

The Executive program is a program that controls which overlay is to be read into core and transfers control to that overlay as required by operator selection. When the overlay program is done, control is relinquished back to the Executive program.

Each overlay contains one or more subroutines.

The IDRS system, including the Executive and all overlays, has been built as a "Task" under the RSX-11D operating system running on the DEC PDP 11/45. IDRS is stored on the RK05 DISK. Overlays are retrieved as needed.

Figure 2-1 describes the overlay structure of IDRS in a "tree" diagram.

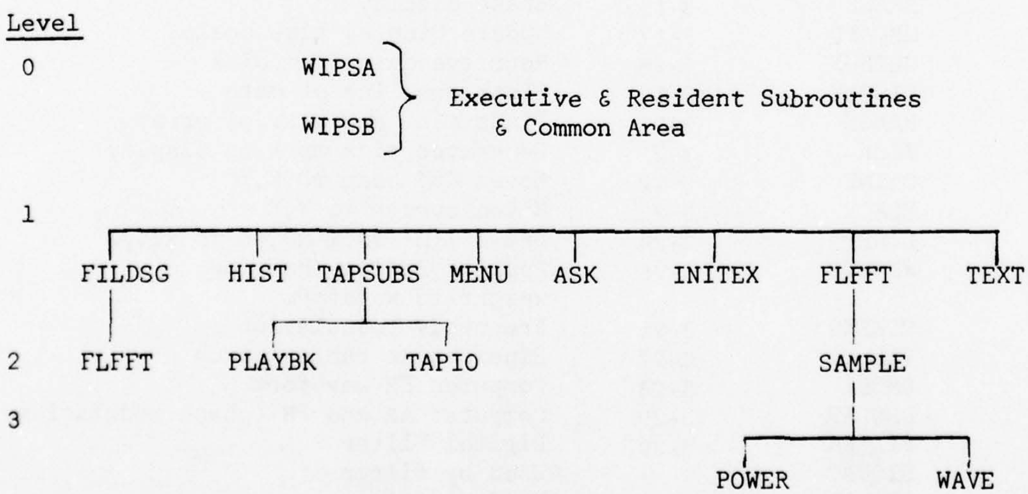


Figure 2-1 IDRS Overlay Structure

As indicated in the diagram, there are four levels of hierarchy. Level 0 contains the resident portion of IDRS. Levels 1, 2 and 3 contain the overlays. All programs in Level 1 share the same core location as do all programs in Levels 2 and 3.

Each program name appearing in Figure 2-1 is actually an RSX source (Fortran) file name. Each source file contains one or more programs or subroutines.

The following sections of this report describe the function of all subroutines contained in each file appearing in Figure 2-1.

As an aid to quick access of the documentation on any subroutine, the following index and brief description has been included:

* These routines require an in-depth knowledge of digital filter design techniques and will not be documented at this time.

FILDSG:	FILDSG	3-31	Design digital filter
	NUMTST		*
	FILDSN		*
	NORMFL		*
	PLYCON		*
	LPLP		*Low pass to low pass transformation
	LPHP		*Low pass to high pass transformation
	LPBE		*Low pass to band elimination transformation
	LPBP		*Low pass to band pass transformation
	WCCONV		*
	CHPOLE		*
	BUPOLE		*
	ARSINH	3-32	Function: hyperbolic arcsine(x)
	COSH	3-33	Function: hyperbolic cosine(x)
	SINH	3-34	Function: hyperbolic sine(x)
	RTSCMP		*
	RENORM		*
	SPOLY		*
	ZPOLY		*
	TAN	3-35	Function: tangent(x)
	FLCHRC	3-36	Plots filter characteristics
FLFFT:	FLFFT	3-37	Fast Fourier Transform
HIST:	HIST	3-40	Calculates a histogram
	HISD	3-41	Displays a histogram
TAPSUBS:	RTAPE	3-42	Reads tape
	WTAPE		Writes tape
	TATT		Attaches logical unit number
	TDET		Detaches logical unit number
PLAYBK:	PLAYBK	3-44	Inputs page of data
TAPIO:	TAPIO	3-45	Input/output tape
	BLDG3	3-46	Reformats header
	DOSTAP	3-47	Reformats header
	TAPEDP	3-48	Saves page of data
MENU:	MENU	3-49	Lists options
ASK:	ASK	3-50	Allows selection of option
	GLOMAX	3-51	Finds global max. and min.
INITEX:	INITEX	3-52	Initializes constants
SAMPLE:	SAMPLE	3-53	Resamples waveform
	XCHGW	3-54	Exchanges Real and Image waveform buffers
	XCHGP	3-55	Exchange position of negative half of power spectrum buffers
	REMOVE	3-56	Remove DC bins from waveform
POWER:	POWER	3-57	Transforms and plots power inputs
WAVE:	WAVE	3-59	Transforms and plots waveform

* These routines require an in-depth knowledge of digital filter design techniques and will not be documented at this time.

TEXT:	POW	3-60	Prints power header
	WAV	3-61	Prints waveform header
	FILTXT	3-62	Prints filter text
HIST2:	EXECUTIVE	3-63	Calculates 2-D histogram
	HISD2	3-64	Displays 2-D histogram
	GETLIN	3-65	

SECTION 3

SOFTWARE DOCUMENTATION

WIPSA

Program Name: EXEC

Purpose: EXEC is the executive module for the Display and Analysis System. The initialization sequence first retrieves the A/D parameters from sector 0 of the data disk, and then calls the ASK overlay, allowing the user to define values for the remaining system parameters from the display keyboard. The control loop enables the user to interactively select and execute various system options from a displayed menu. Two types of options are available during system operation. The more complex options require overlay swapping while the simpler ones allow the user to reset system parameters on line. These system parameters can be switched to determine if a transform subroutine is to be turned on or not.

Labeled Common:

/BUFFER/BUF (2049) -

The BUFFER common block is the data line buffer; when processing or displaying data, this buffer contains one data line at a time. When not being utilized as a line buffer, it serves as a temporary message area by the Tektronix I/O routines.

/DUF/IBUF (1280) -

The DUF common block is the general integer buffer used in data retrieval from disk.

/BUFF2/BUF2 (1024) -

The BUFF2 common area is the auxiliary floating point data buffer that is used for resampling and buffer interchanges.

/SAVE/ S1(10,2), S2(10,2), S3(10,2) S4(10,2), S5, S6, INDEX

This common area is used to save filter's initial conditions and phase conditions necessary to maintain continuity between data pages.

/FLPARM/ A1(10), B1(10), A2(10), B2(10), A3(10), B3(10), KPB,
A4(10), B4(10), AK(10), Y1(10,2), Y2(10,2), Y3(10,2),
Y4(10,2), NSECTN

This common block contains filter parameters and filter initial conditions.

/AUTO/ IBYP, NCM, NF, NFILE(99), LTAP, IAUTOP, INMOP, MOP(300), MCOP(300), INOP, IA, IPOIN, ITM(4)

This common block contains parameters which allow the automatic sequencing through a specified sequence of transformations.

/TCOSW/TCOS(512), NPTOLD

This common area contains a sine/cosine table which is used by the weighting routines.

<u>Unlabeled Common:</u>

NC	-	Number of digitized channels
ISR	-	Sampling rate (double integer)
IH	-	Digitization start time (hours)
IM	-	Digitization start time (minutes)
IS	-	Digitization start time (seconds)
IEH	-	Digitization stop time (hours)
IEM	-	Digitization stop time (minutes)
IES	-	Digitization stop time (seconds)
ISECS	-	Total run time (seconds)
NWLB	-	Number of words in last block (1 block = 8192 words)
NOB	-	Number of blocks
ISTAT	-	Run number
IFLAG	-	Not used
IBSOLB	-	Beginning sector of last block to disk
IASLB	-	Actual block size of last block
ILS	-	Last sector containing digitized data (double integer)
ISECT	-	Current sector (double integer)
LOCAL	-	Local/global switch (0/1)
LPTS	-	Number of waveform points to overlap consecutive lines
NPTS	-	Number of points per line
KPTS	-	Number of points in power
ITH	-	Initial spectral index to be displayed
JTH	-	Final spectral index to be displayed
NLINES	-	Number of lines per page
ISK	-	Skip point index
IC	-	Selected channel number
KTH	-	Not used
LOCALG	-	Global wave flag
IDL	-	Details line flag
GMAX	-	Global wave max
GMIN	-	Global wave min
IHAN	-	Weighting flag
IREC	-	Rectification flag
ISTW	-	Store wave flag
ISTP	-	Store power flag
IFILI	-	Filter one flag

IFG	-	Frequency convert flag
MOD	-	Buffer modification flag
IFF	-	Filter initialization flag
IDM	-	Demodulation flag
IDIS	-	Display mode flag
IOVRLY	-	Overlay number
ISTAT1	-	Display status wave
ISTAT2	-	Display status power
ISTAT3	-	Display status not assigned
IAUP	-	Power average flag
IWO	-	Wave display flag
IPO	-	Power display flag
ITO	-	Display flag
IFILE	-	Tape roll in/out flag
IPLOG	-	Log of power flag
IFLAG1	-	System flag
IFLAG2	-	System flag
JUMP	-	System flag
ITAPE	-	Tape roll in/out flag
IASKIT	-	System EXEC flag
IOTEMP	-	Temp overlay number
JTAPE	-	Tape roll in/out flag
PHO	-	Phase (for frequency conversion)
FRSHFT	-	Frequency shift
IDR	-	Phase derivative flag
ISTEXT(20)	-	Status text
IFIL2	-	Filter two flag
IRSAM	-	Resample index
PHIO	-	Initial phase for (DMODF)
IDC	-	DC removal flag
SMPLEF	-	Effective sample rate
ISWIT	-	System flag
IAXIS	-	Waveform zero-line flag
NPTS2	-	2 x no. points
NPTS4	-	4 x no. points
NPT	-	No. of displayed points
ITH1	-	Temporary ITH point
JTH1	-	Temporary JTH point
IT(1)	-	Current time index (hours; double integer)
IT(2)	-	Current time index (minutes; double integer)
IT(3)	-	Current time index (seconds; double integer)
IT(4)	-	Current time index (number of sample points beyond current second; double integer)
TIMAX1	-	Max value buffer 1 (Display 1)
TIMIN1	-	Min value buffer 1 (Display 1)
TIMAX2	-	Max value buffer 2 (Display 2)
TIMIN2	-	Min value buffer 2 (Display 2)

TIMAX3	-	Max value power buffer
TIMIN3	-	Min value power buffer
JT(1)	-	Temporary time index (hours; double integer)
JT(2)	-	Temporary time index (minutes; double integer)
JT(3)	-	Temporary time index (seconds; double integer)
JT(4)	-	Temporary time index (number of sample points beyond temporary second; double integer)
T2MAX1	-	Temporary TIMAX1
T2MIN1	-	Temporary TIMIN1
T2MAX2	-	Temporary TIMAX2
T2MIN2	-	Temporary TIMIN2
T2MAX3	-	Temporary TIMAX3
T2MIN3	-	Temporary TIMIN3

File: WIPSA

Program Name: DISKF

Function: Performs unformatted disk input and output. Device can be either RP04 or RK05 disks. Disks must be mounted as foreign.

Calling Sequence: CALL DISKF (FUN, B, L, I)

Subroutine Parameters:

FUN (Real) - Disk function read/write
B(Real) - Buffer address
L (Integer) - Number of words (10 bit)
I (Double Integer) - Starting sector number

File: WIPSA

Program Name: READPK

Function: Transfers data from RP04 disk to core; user specifies buffer address, word count and starting sector number.

Calling Sequence: CALL READPK (B, L, I)

Subroutine Parameters:

B (Real) - Buffer address
L (Integer) - Word count
I (Double Integer) - Starting sector number

File: WIPSA

Program Name: WRITPK

Function: Transfer data from core to RP04 disk, user specifies buffer address, word count and starting sector number.

Calling Sequence: CALL WRITPK (B, L, I)

Subroutine Parameters:

- B (Real) - Buffer address
- L (Integer) - Word count
- I (Double Integer) - Starting sector number

Common Area: None required

File: WIPSA

Program Name: OUTA

Function: OUTA is used to output a string of text characters to the Tektronix storage tube. Text information can be placed anywhere on the screen by first calling subroutine PLACE. The string of text can contain two special control characters; > or< will cause a line feed and carriage return to occur. Repeated occurence of a control character will cause repeated occurrence of its function.

Calling Sequence: CALL OUTA (N, STRING)

Subroutine Parameters:

N (Integer)	-	Number of characters to be output. This count should also include control characters.
String	-	Buffer containing ASCII characters (2 characters per word).

Common Area: None required

Note: Single quotation marks can be used in place of an array name. For example:

```
Call OUTA (N, 'TEXT')
```

File: WIPSA

Program Name: CHOUT

Function: This routine is used to output a single character to the Tektronix display.

Calling Sequence: CALL CHOUT (J)

Subroutine Parameters:

J - The ASCII value of a character to be output to the Tektronix display.

Common Area: None required

File: WIPSA

Program Name: INPUTI

Function: This routine is used to input integer numbers via the storage tube keyboard. If more than one number, the numbers must be separated by commas.

Calling Sequence: CALL INPUTI (N, ICHAR, IRAY)

Subroutine Parameters:

N	-	Number of total characters expected to be typed in. (commas included)
ICHAR (Integer)	-	Array to receive characters typed in, ICHAR must be dimensioned large enough to contain the number of characters expected to be typed in. (2 characters per word)
IRAY (Integer)	-	Integer array to receive binary values of numbers typed in.

Common Area: None required

File: WIPSB

Program Name: RECT

Function: Performs rectification of waveform data in a number of ways.

Calling Sequence: CALL RECT (NPTS, IREC)

Subject Parameters:

NPTS (Integer) -	Number of points in data buffer
IREC (INTEGER) =	1 - Top rectification
	2 - Bottom rectification
	3 - Full wave rectification

Common Area: Resident /BUFFER/ - Data buffer

File: WIPSB

Program Name: SAVERF

Function: Saves initial conditions of filter and phase parameters used in demodulation to maintain continuity between data pages.

Calling Sequence: CALL SAVERF (PHO, PHIO, ISTAT1, IFF)

Subject Parameters:

PHO (Real) - Phase used in frequency translation.

PHIO (Real) - Phase of last part of last line needed for DMODF.

ISTAT1 (Integer) - Status of waveform buffer

IFF (Integer) - No filter/Filter flag

Common Area:

Resident: /FLPARM/

Resident: /SAVE/ FLPARM saved here

File: WIPSB

Program Name: GETLOG

Function: This routine takes the log base 10 of a power spectrum and multiplies the result by 10.

Calling Sequence: CALL GETLOG (KPTS, AMIN)

Subject Parameters:
- KPTS (Integer) - Number of points
- AMIN (Real) - Smallest number encountered (db) X (-1)

Common Area:
- /BUFFER/ - Data buffer

File: WIPSB

Program Name: SUMBUF

Function: This subroutine will sum the normalized data buffer to the auxiliary buffer for n times. When JJ ≠ 0 (or = n) the auxiliary buffer is put in the general data buffer.

Calling Sequence: CALL SUMBUF (KPTS, JJ)

Subject Parameters:
KPTS (Integer) - Number of points
JJ (Integer) - 0; sum from general buffer to auxiliary buffer
n; sum to auxiliary buffer and transfer to general buffer

Common area:
/BUFF2/ - Auxiliary data buffer
/BUFFER/ - General data buffer

File: WIPSB

Program Name: ERASE

Function: Erases the Tektronix display and places cursor in the upper left corner.

Calling Sequence: CALL ERASE

Subject Parameters: None required

Common Area: None required

File: WIPSB

Program Name: UPDATE

Function: This routine will update the wave beginning time by JPTS. It also adjusts for overlap, skip, and sample rate.

Calling Sequence: CALL UPDATE (JTIME, JPTS, LPTS, ISK, ISR)

Subject Parameters:

JTIME(1) (Double Integer) - Start time in hours
JTIME(2) (Double Integer) - Start time in minutes
JTIME(3) (Double Integer) - Start time in seconds
JTIME(4) (Double Integer) - Start time in points
JPTS (Integer) - Number of points
LPTS (Integer) - Number of overlapping points
ISK (Integer) - Number of skip points
ISR (Double Integer) - Sample rate

Common Area: None required

File: WIPSB

Program Name: GETBUF

Function: This routine accesses the digitized raw data stream on the RPO4 disk via time code, converts the data samples to floating point and transfers them to the labeled common BUFFER block.

Calling Sequence: CALL GETBUF (NUM, IEF, IRS, INP, IT, ILS, NC, IC, TSR, IH, IM, IS, TSK)

Subject Parameters:

NUM - Number of data samples to be retrieved from the RPO4
IEF - Retrieval error flag
  No errors = 0
  Requested Time Code out of bounds ≠ 0
IRS - Number of five-sector blocks from end of raw data file
INP - Number of data samples actually transferred without error
IT(1) (Double Integer) - Begin hour
IT(2) (Double Integer) - Begin minute
IT(3) (Double Integer) - Begin second
IT(4) (Double Integer) - Begin point
ILS (Double Integer) - Last sector used
NC (Integer) - Number channels
IC (Integer) - Channel wanted
ISR (Double Integer) - Sample rate
IH (Integer) - Waveform begin hour
IM (Integer) - Waveform begin minute
IS (Integer) - Waveform begin second
ISK (Integer) - Skip count

Common Area:

/DUF/ Input raw data here
/BUFFER/ Floating point data here

File: WIPSB

Program Name: DISPLY

Function: This routine plots multiple waveforms or power spectra lines on the Tektronix screen with upper and lower reference axes and optional zero axis.

Calling Sequence: CALL DISPLY (MODE, NLINES, ILINE, ITH, JTH, YMAX, YMIN, IRL, IPLOG)

Subject Parameters:

MODE (Integer) - Waveform/spectrum switch
Waveform = 0

NLINES (Integer) - Number of lines
LINE (Integer) - Current line number
($1 \leq$ Line # $\leq$ NLINES)
ITH (Integer) - Initial spectral index to be displayed
JTH (Integer) - Last spectral index to be displayed
(The ITH and JTH indices are used to expand or zoom into spectral bands)
YMAX (Real) - Maximum Y value to be plotted on current line
YMIN (Real) - Minimum Y value to be plotted on current line
IAXIS (Integer) - Zero axis switch
Draw axis $\neq$ 0
Omit axis = 0
IPLOG - DB log scaling, IPLOG = 0 No log
= DB of scaling

Common Area:

/BUFFER/ - Data to be displayed

File: WIPSB

Program Name: RANGE

Function: Finds maximum and minimum of BUF between indices ITH and JTH.

Calling Sequence: CALL RANGE (BUF, ITH, JTH, FMAX, FMIN)

Subroutine Parameters:

- BUF (Real) - Input buffer array
- ITH (Integer) - Starting index > 0
- JTH (Integer) - End index ≤ 1024
- FMAX (Real) - BUF maximum value
- FMIN (Real) - BUF minimum value

Common Area: None required

File: WIPSB

Program Name: TICK

Function: This routine draws a horizontal reference axis with vertical tick marks on the Tektronics display.

Calling Sequence: CALL TICK (JY, JX1, JX2, NT, JH)

Subroutine Parameters:

JY (Integer) - Y value for reference axis
JX1 (Integer) - Initial X value for reference axis
JX2 (Integer) - Last X value for reference axis
NT (Integer) - Number of internal tick marks. Tick marks at the ends of the reference axis are not included.
JH (Integer) - Tick mark length in screen units. Positive number places tick marks above reference axis and negative places them below.

Common Area: None required

File: WIPSB

Program Name: CLINE

Function: This routine plots piecewise continuous line segments by successive calls. The initial call is made with the pen up. After this, the subsequent calls are made with the pen down. A call to this routine with the pen up sets the Tektronix into the Graphic Linear Interpolation Mode.

Calling Sequence: CALL CLINE (IX, IY, IPEN)

Subroutine Parameters:

IX (Integer) - An x value, expressed in screen units (0 to 1023)
IY (Integer) - A Y value, expressed in screen units (0 to 760)
IPEN (Integer) - Pen up/down switch
0 = pen up
1 = pen down

Common Area: None required

File: WIPSB

Program Name: PLACE

Function: This routine places the current graphics position or alpha cursor at the specified x - y coordinates of the display screen. This routine leaves the Tektronix display in the alpha mode.

Calling Sequence: CALL PLACE (JX, JY)

Subroutine Parameters:

JX (Integer) - Specified x coordinate
JY (Integer) - Specified y coordinate

Common Area: None required

File: WIPSB

Program Name: LINE

Function: This routine draws a line segment from point x, y to point x, y on the Tektronix display screen.

Calling Sequence: CALL LINE (IX1, IY1, IX2, IY2)

Subroutine Parameters:

IX1 (Integer) - A value for x, expressed in screen units (0 to 1023)
IY1 (Integer) - A value for y, expressed in screen units (0 to 760)
IX2 (Integer) - A value for x, expressed in screen units (0 to 1023)
IY2 (Integer) - A value for y, expressed in screen units (p to 760)

Common Area: None required

File: WIPSB

Program Name: WEGHTG

Function: Applies Hanning or Hamming weighting to waveform x(n) in /Buffer/. The weighting function w(n) is defined as:

Hanning:

$$w(n) = \frac{1}{2}\left[1 - \cos \frac{2\pi n}{N-1}\right], \quad 0 \leq n \leq N - 1$$

Hamming:

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}, \quad 0 \leq n \leq N - 1$$

The output waveform y(n) = w(n)x(n) is placed back into /Buffer/.

Calling Sequence: CALL WEGHTG (NPOINT, NFLAG, MODE)

Subroutine Parameters:

NPOINT (Integer) - Number of points
NFLAG (Integer) - 1; Hanning
2; Hamming

MODE (Integer) - 1; Real Data
2; Complex Data

Common Area:

/BUFFER/ - Data buffer
/TCOSW/ - Half-wave cosine for lookup

Note: Routine should not be called for NPOINT < 2.

File: WIPSB

Program Name: CSMIXG

Function: Perform frequency translation or shift by multiplying a real input sequence x(n) by $e^{\frac{-j2\pi f_c n}{f_s}}$ where $f_c \equiv$ shift frequency and $f_s \equiv$ sampling rate. The complex result is placed into /Buffer/. A sine and cosine table lookup is used.

Calling Sequence: CALL CSMIXG (PHO, FRSHFT, SMPLEF, NPOINT)

Subroutine Parameter:

PHO (Real)	- Phase or last argument used for last point of last line
FRSHFT (Real)	- Frequency shift
SMPLEF (Real)	- Sampling frequency
NPOINT (Integer)	- Number of points

Common Area:

/Buffer/ - Input/output data buffer

File: WIPSB

Program Name: VAL

Function: Perform table lookup of sine and cosine values from a quarter-wave sine table stored in FLFFT.

Calling Sequence: CALL VAL (X, Y, I)

Subroutine Parameters:

X (Real)   - Sine
Y (Real)   - Cosine
I (Integer) - Index into table $1 \leq I \leq 1024$

Common Area:

/SINCOS/ (Real) - Quarter-wave sine table found in FLFFT

File: WIPSB

Program Name: DMODF

Function: This subroutine computes the derivative of the phase in EMAG. The resultant is stored back in EMAG. PHI0 defines the phase of the 0-th point. This is included to give the flexibility of putting together several sections of a long waveform.

Calling Sequence: CALL DMODF (NPT, PHI0)

Subroutine Parameters:

NPT (Integer) - Number of points
PHI0 (Real) - Phase of 0-th point

Common Area: /Buffer/ Real (1024), EMAG (1024)

File: WIPSB

Program Name: DMODAP

Function: Computes magnitude (AM waveform) and phase waveforms from the In-phase and Quadrature waveform contained in the Real and EMAG arrays respectively.

Calling Sequence: CALL DMODAP (NPOINT)

Subroutine Parameters:

NPOINT (Integer) - Number of points

Common Area:

/Buffer/ Real (1024), EMAG
Real contains the I - waveform
EMAG contains the Q - waveform

File: WIPSB

Program Name: FILTER

Function: This routine will filter a sequence of data points.

Calling Sequence: CALL FILTER (NPOINT, NCLFLG, MODE)

Subroutine Parameters:

NPOINT - Number of points in the sequence
NCLFLG - = 0 set initial conditions to zero
= 1 use previous conditions
MODE - Will break the buffer up into either one or two sequences
Mode = 1 for one sequence that can go from 1 to 204
Mode = 2 for two sequences that can go from 1 to 1024 and from 1024 to 2048

Common Area:

/Buffer/ (Real) - Input/output buffer
/FLPARM/ - Coefficients of the biquadratic function:

$$AK * \frac{(1 + B17^{-1} + B27^{-2})}{(1 - A17^{-1} - A27^{-2})}$$

WSECTN = Number of biquadrature sections in the filter

File: FILDSG

Program Name: FILDSG

Function: FILDSG allows the user to interactively design filters to be used by the IDRS system. The filter classes that can be picked are Chebyshev filter and Butterworth filter. The pass band characteristics that may be chosen are: low pass, high pass, band pass, and band elimination. The maximum number of poles allowed is 18.

Once the filter has been designed, the user can display the following characteristics: impulse response, amplitude response, phase response, group delay, and step response. Also, the user can get a hardcopy of the filter parameters. (See Appendix A for a technical description of the filter design logic.)

Calling Sequence: CALL FILDSG

Common Area: Same as EXEC

Notes: Due to the technical nature of this program, several special purpose subroutines were developed. Some of these are only usable within the context of the filter design program. Any documentation of these routines would require an in-depth knowledge of digital filter design techniques. Accordingly, only those routines which are general enough to permit usage outside of the context of FILDSG will be documented.

The following routines will not be documented:

NUMTST
FILDSN
NORMFL
PLYCON
LPLP
LPHP
LPBE
LPBP
WCCONV
CHPOLE
BUPOLE
RTSCMP
RENORM
SPOLY
ZPOLY

File: FILDSG

Program Name: ARSINH

Function: Computes the inverse hyperbolic sine of x in the following manner:

$$y = \mathrm{Log}_{10} \left(x + \sqrt{x^2 + 1}\right)$$

The result is retrieved in x.

Calling Sequence: (Function) Y = ARSINH(X)

Subroutine Parameters: X(Real) - Input/output value

Common Area: None

File: FILDSG

Program Name: COSH

Function: Computes the hyperbolic cosine of X in the following manner:

$$Y = \frac{e^{x} + e^{-x}}{2}$$

The result is returned in X.

Calling Sequence: (Function) Y = COSH(X)

Subroutine Parameters: X(Real) - Input/output value

Common Area: None required

AD-A053 684 PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y F/G 9/4
INTERACTIVE DIGITAL RECEIVER SIMULATOR (IDRS) SYSTEM USER'S MAN--ETC(U)
MAR 78 R E BOZEK, J C BREDA, O P GUPTA F30602-76-C-0126
UNCLASSIFIED PAR-76-30 RADC-TR-78-44 NL

2 OF 2
AD
A053684

END
DATE
FILMED
6 -78
DDC

File: FILDSG

Program Name: SINH

Function: Computes the hyperbolic sine of X in the following manner:

$$Y = \frac{e^{x} - e^{-y}}{2}$$

The result is returned in X.

Calling Sequence: (Function) Y = SINH(X)

Subroutine Parameters: X(Real) - Input/output value

Common Area: None required

File: FILDSG

Program Name: TAN

Function: Computes the tangent of X in the following manner:

$$Y = \frac{\text{sine}(X)}{\text{cosine }(X)}$$

The result is returned in X.

Calling Sequence: (Function) Y = TAN(X)

Subroutine Parameters: X(Real) - Input/output value

Common Area: None required

File: FILDSG

Program Name: FLCHRC

Function: Generates plots of the filter characteristics for the filter previously designed. Plots include the following:

Impulse response
Log amplitude response
Phase response
Group delay
Step response.

The coefficients of the transfer function (filter) are also output so that a digital filter may be realized outside of IDRS.

Calling Sequence: CALL FLCHRC(NPOLE1, KPB1, KTYPE, FRACW1, FRACW2, EPSLON, NWCL, NWCH)

Subroutine Parameters:

NPOLE1 (Integer)	- Number of poles
KPB1 (Integer)	- Filter type 0; Chebyshev 1; Butterworth
KTYPE (Integer)	- Filter type (low pass, high pass, band pass or band elimination)
FRACW1 (Real)	- High pass cutoff as fraction of sampling frequency
FRACW2 (Real)	- Low pass cutoff as fraction of sampling frequency
EPSLON (Real)	- Pass band ripple (db)
NWCL, NWCH	- Frequency limits in Hz

File: FLFFT

Program Name: FLFFT

Function: FLFFT is a subroutine written in PDP 11/45 assembly language for performing a forward or inverse Fast Fourier Transform or power spectrum of N complex data points, where N is a power of 2 in the range ($8 \leq N \leq 1024$). Since FLFFT was written to take advantage of the PDP 11/45's floating point processor, all input and output data points are assumed to be in the single precision floating point format. (2 words/value).

The forward transform is defined by

$$A_r = \sum_{k=0}^{N-1} X_k \exp(-2\pi jrk/N), \quad r = 0, \ldots, N-1 \qquad (1)$$

Where Ar is the rth coefficient of the FFT and X is the Kth complex sample of a time series which consists of N samples.

The inverse can be written as

$$X_r = 1/N \sum_{k=0}^{N-1} A_k \exp(2\pi jrk/N), \quad r = 0, \ldots, N-1 \qquad (2)$$

For both the forward and inverse transforms, FLFFT stores the transform points in two arrays. The real values of a transform are stored in REAL and the corresponding imaginary values in EMAG. The results of the power spectrum are stored in REAL. In calculating the inverse, FLFFT does not divide the values by N as in (2) above. This must be done by the user if he desires actual values rather than relative magnitudes.

Calling Sequence: Before calling FLFFT the data points must be loaded into the common area labled Buffer. REAL values are entered into array REAL, imaginary values into EMAG, starting at the top of each array respectively. The FORTRAN statements required to call FLFFT are the following:

```
COMMON/BUFFER/REAL (1024), EMAG (1024),
FMAX
 .                 .
 .                 .
 .                 .
CALL FLFFT (NPTS, INVRS, IPOWER, IMAG)
```

Upon return from FLFFT, the real parts of the transformed points will be in the array REAL, the imaginary parts in EMAG. If the power spectrum has been requested, the power spectral estimates will be in REAL.

FLFFT (Continued)

Subroutine Parameters:

NPTS (Integer)	-	The number of data points must be a power of two and not less than 8 or greater than 1024.
INVRS (Integer)	-	The forward/inverse flag is set to 0 to request a forward spectrum is to be calculated, and is set to 1 if not.
IPOWER (Integer)	-	The power spectrum option flag is set to 0 if the power transformation, and set to 1 for an inverse transformation.
IMAG (Integer)	-	The real/imaginary data flag is set to 0 if the input data is strictly real, and is set to 1 if it is complex. Setting the flag to zero instructs FLFFT to clear the imaginary data array EMAG before calculating the FFT.

Common Area: /Buffer/ Real (1024), EMAG (1024), E

Frequencies of Power Spectral Estimates:

When the data being transformed is real, N/2+1 non-redundant spectral estimates are obtained, and each of these estimates corresponds to a frequency in the input data. In terms of the number of data points, the ith spectral estimate represents a frequency of i cycles per N data points. The range of i is 0, 1, ..., N/2. For 1024 points, for example, 513 cycles per 1024 data points, in steps of one cycle.

To convert "cycles per N data points" to units of frequency of the input data wave, one merely substitutes the time required to sample the N data points for the phrase "per N data points". Thus, if the 1024 data points in the above examples had been sampled in 12 seconds, the frequencies of the estimates would be 1/12 of a cycle per second (one cycle per 12 seconds) to 42 2/3 cycles per second.

If the power spectrum calculation has been requested, the spectral estimates are placed in the upper N/2+1 positions of the REAL array, and the previous contents of those positions are lost. If the user wishes to calculate the phase angles of the frequency components, he must not request the power spect um calculation. The phase angles of the frequency components are then given as

$$\Theta_i = \mathrm{Tan}^{-1} \frac{\mathrm{EMAG(i)}}{\mathrm{REAL(1)}}$$

The sampling of a source wave must be at a rate of at least twice the highest frequency contained in the source. This rate avoids aliasing, or confusion of higher frequencies for lower ones.

FLFFT (Continued)

Programming Techniques:

The following techniques were used to minimize execution time and storage requirements.

(1) The transform itself is done by the decimation in frequency technique described by Cochran, W.T., et al in a paper titled, "What is the Fast Fourier Transform?" Use of this method minimizes "bit reversal".

(2) No multiplications or sine/cosine table look-up are required for the final two column transformations, hence these transformations are done in two special loops, separately from the main loop.

(3) A quarter wave (256 value or 512 location) sine/cosine table is used.

File: HIST

Program Name: HIST

Function: Computes a one-dimensional histogram of any waveform or power data.

Calling Sequence: CALL HIST(NLINES, ILS)

Subroutine Parameters:

NLINES - Number of lines of data to be used in the histogram
ILS - The number of the last sector of data

Common Area: BUFFER

File: HIST

Program Name: HISD

Function: Displays a one-dimensional histogram of any waveform or power data.

Calling Sequence: CALL HISD(NB, H, HMAX, L, K)

Subroutine Parameters:

- NB - Number of bins in the histogram
- H - The array that contains the already scaled histogram values
- HMAX - The maximum value of the components of H
- L - Number/division on the vertical scale of the histogram
- K - Number of divisions on the vertical scale of the histogram

Common Area: None

File: TAPSUBS

RSX-11D Non-File Structured Magnetic Tape Input/Output Routines:

```
Call      TATT (IERR, LUN, IUNIT)
          Assigns a logical unit number to a magtape unit, and attaches
          the device to the task.  Also sets tape characteristics to
          nine-track, odd parity.
Call      TDET (IERR, LUN)
          Detaches the magtape from the task and releases the logical
          unit number.
Call      RTAPE (IERR, LUN, IBUFF, ICOUNT, IREAD)
          Reads one record from magtape.
Call      WTAPE (IERR, LUN, IBUFF, ICOUNT)
          Writes one record to magtape.
Call      WEOF (IERR, LUN)
          Writes one end-of-file mark.
Call      REWIND (IERR, LUN)
          Rewinds magtape.
Call      REWOFF (IERR, LUN)
          Rewinds magtape and places off-line.
Call      TSPACE (IERR, LUN, N)
          Spaces past records on magtape either direction.
Call      TFILE (IERR, LUN, M)
          Spaces past file on magtape either direction.
Call      TSET (IERR, LUN, K)
          Sets tape characteristics.
Call      TEND (IERR, LUN)
          Searches for double end-of-file mark.
```

Arguments:

```
IERR = error code
+ 1 -- successful operation
- 1 -- eof encountered
- 2 -- eot encountered
- 3 -- bad parameter list
- 4 -- hardware fatal error
- 5 -- logical unit number not assigned to magtape unit
- 6 -- logical unit number previously assigned to magtape unit
- 7 -- magtape unit number previously assigned to logical unit number
- 8 -- other unfriendly error (reported by message output handler)

LUN    = logical unit number
IUNIT  = physical unit number
IBUFF  = data buffer address
ICOUNT = count in bytes to read or write
IREAD  = bytes actually read
N      = number of records to space (signed)
```

M = number of files to space (signed)

K = characteristic word (see RSX-11D Device Handler Reference Manual, Chapter 7, pages 5 through 9.)

- All variables are one word integers.
- Programs are contained in module TAPSOB.OBJ

File: PLAYBK

Program Name: PLAYBK

Function: Inputs a "PAGE" of data previously stored with the (TP) option. This option currently exists only at the RADC Building 3 facility.

Calling Sequence: CALL PLAYBK

Subject Parameters: None

Common Area: EXEC unlabled common

File: TAPIO

Program Name: TAPIO

Function: This overlay performs the roll in and roll out options. Roll out will copy the current digitized data stream from the RP04 disk onto mag tape; whereas, roll in will read a previously rolled out data stream from mag tape back onto the system data disk. The value of the system parameter JTAPE will determine which of these options is performed. The data from the disk is copied onto mag tape starting at sector 0 and each tape record will contain eight sectors.

Calling Sequence: CALL TAPIO

Parameters: JTAPE (Integer) - Roll in/roll out switch
Roll in = 0
Roll out $\neq$ 0

Subroutine Parameters: None

Common Area: Same as EXEC

File: TAPIO

Program Name: BLDG3

Function: This routine makes tapes from Building 3 RADC compatible with PAR IDRS system. It acts as interface between the header information and the RP02 disk at Building 3 and the RP04 at PAR.

Calling Sequence: CALL BLDG3

Subroutine Parameters: None

Common Area:

All unlabled common as in EXEC
/DUF/ - Input/output buffer

File: TAPIO

Program Name: DOSTAP

Function: This routine makes the digital tapes generated at PAR's A/D facility running under the DOS system compatible with IDRS which runs under the RSX operating system.

Calling Sequence: CALL DOSTAP

Subroutine Parameters: None

Common Area:

All unlabled common as in EXEC and
/DUF/ - Input/output buffer

File: TAPIO

Program Name: TAPEDP

Function: This routine enables the user to store several "pages" of transformed data into magnetic tape for subsequent input to IDRS via the "TI" option.

Calling Sequence: CALL TAPEDP

Subroutine Parameters: None

Common Area: Same as EXEC

Note: TAPEDP uses RP04 disk storage beginning at ILS + 9997 to build tape image file before transferring to tape.

File: MENU

Program Name: MENU

Function: This overlay displays the list of the Display and Analysis Subsystem options and their character codes on the Tektronix storage tube.

Calling Sequence: CALL MENU

Subroutine Parameters: None

Common Area: None

File: ASK

Program Name: ASK

Function: The ASK overlay handles the interactive I/O between the user and the computer. The user, by inputting one or two character commands, can activate various parameter switches which will turn on certain portions of the system program. ASK insures that all commands/switch parameters are legal.

Calling Sequence: CALL ASK

Subroutine Parameters: None

Common Area: Same as EXEC

File: ASK

Program Name: GLOMAX

Function: Computes minimum and maximum values over entire waveform file.

Calling Sequence: CALL GLOMAX

Subroutine Parameters: None

Common Area:

All unlabled common same as in EXEC and /DUF/
GMAX (Real) - Waveform maximum
GMIN (Real) - Waveform minimum

File: INITEX

Program Name: INITEX

Function: INITEX is the first overlay that IDRS calls, and is usually only called once, each time IDRS is run. INITEX initializes all parameters to a starting value for the IDRS system. This insures that IDRS is at the same starting point each time it is run.

Calling Sequence: CALL INITEX

Subroutine Parameters: None

Common Area: Same as EXEC

File: SAMPLE

Program Name: SAMPLE

Function: This routine will take every X point (resample) from the real and imag array (buffer) and place resampled data in auxiliary buffers until they are filled up. When the auxiliary buffers are filled, the resampled data is placed into real and imag buffers.

Calling Sequence: CALL SAMPLE(IRSAM, INUM, INDEX, NPTS, JJ)

Subroutine Parameters:

IRSAM (Integer)	- The resample rate
INUM (Integer)	- This parameter is originally set to zero, then it is increased by one internally on each call to SAMPLE. When IRSAM = INUM then the auxiliary buffers are put into the real and imag buffer as resampled data
INDEX (Integer)	- Originally set to 1 on each resample loop
NPTS (Integer)	- Number of points in buffer
JJ (Integer)	- Index for auxiliary buffer

Common Area: /Buffer/ - Resampled data here

File: SAMPLE

Program Name: XCHGW

Function: This routine will exchange the real and imaginary buffers for display purposes.

Calling Sequence: CALL XCHGW(NPTS)

Parameters:

NPTS (Integer) - Number of points in array

Common Area: /Buffer/ (Real) Real(1024), EMAG(1024) - Input/output buffer

File: SAMPLE

Program Name: XCHGP

Function: This routine exchanges the positive half of the power spectrum frequency with the negative half for display purposes.

Calling Sequence: CALL XCHGP(NPTS)

Subroutine Parameters:

NPTS (Integer) - Number of points in power spectrum array

Common Area:

/Buffer/ (Real) - Input/output buffer

File: SAMPLE

Program Name: REMOVE

Function: This routine will remove the DC bias of a data sequence.

Calling Sequence: CALL REMOVE(NPTS, MOD)

Subroutine Parameters:

NPTS (Integer) - Number of points in the sequence

MOD (Integer) - Will break the buffer up into either one or two sequences for DC removal
MOD = 1 For one sequence that goes from 1 to 2048
MOD = 2 For two sequences that can go from 1 to 1024 and from 1025 to 2048.

Common Area:

/Buffer/ (Real) - Input/output buffer

File: POWER

Program Name: POWER

Function: The main functions of POWER are to access the long waveform data via time code or the appropriate line buffers; calculate the power spectrum, and display the data line by line on the storage tube screen.

The long waveform data can be found in one of three areas on the RP04 disk. The IDRS data disk format is shown below. The first type is the Digitized Data Sample File which is created by the A/D Conversion Subsystem and accessed by POWER each time it displays a page of new data. This data area starts on sector 1 of the RP04 and continues for n sectors. The number of sectors, n, is dependent on the time bandwidth product of the digitized signal. As waveform data is extracted from this file for display, a copy of each data line is saved in the Waveform Line Buffers. Saving a copy of the data page line by line reduces the overhead required to re-display the same page with local or page scaling. When power spectra are calculated, they are also stored on a line by line basis in the Power Spectrum Line Buffers. Having both the current page waveform and spectrum lines stored on the disk reduces the amount of computation required to switch back and forth between local or page scaling and waveform or power spectra displays.

When creating a new waveform or spectra display page, the minimum and maximum values for each line are found and stored on the disk with each data line's starting time code. In local scaling, each line is scaled to its own minimum and maximum, while, page scaling uses the page minimum and maximum to determine the scaling for each line. The parameter, ISWIT, is used in global scaling to enable POWER to retrieve all the page data and determine the page minimum and maximum before any lines of data are displayed.

The details option enables a user to zoom in and expand a band of frequencies of a spectra page across the display screen. The minimum and maximum values for each line may change with this option, therefore, the minimum and maximum values within the frequency band to be expanded have to be determined each time a page is displayed or re-displayed. Details utilizes the parameters, ITH and JTH, which are the indices for the first and last power terms in the frequency band of interest. If ITH equals 1 and JTH equals NPTS, the Details option is off and the complete spectrum will be displayed. The power profile option is set when the parameter IFILE is not zero. When this option is selected, the sum of the power from the ITH to the JTH terms for each line of a spectra display is stored on the data disk in the sector after the last display line buffer. After the spectra display page is completed, the user has to hit or to display the power profile plot. Each

POWER (Continued)

point in the profile represents a line in the spectra display and is the square root of the average power value for the frequency band of interest.

Subroutine Parameters: None

Common Area: Same as EXEC

File: WAVE

Program Name: WAVE

Function: The main functions of WAVE are to access the long waveform data via time code or the appropriate line buffers, and display the data line by line on the storage tube screen.

The long waveform data can be found in one of two areas on the RP04 disk. The IDRS data disk fromat is shown below. The first type is the Digitized Data Sample File which is created by the A/D Conversion Subsystem and accessed by WAVE each time it displays a page of new data. This data area starts on sector 1 of the RP04 and continues for n sectors. The number of sectors, n, is dependent on the time bandwidth product of the digitized signal. As waveform data is extracted from this file for display, a copy of each data line is saved in the Waveform Line Buffers. Saving a copy of the data page line by line reduces the overhead required to re-display the same page with local or page scaling. Saving the current page waveform on the disk reduces the amount of computation required to switch back and forth between local or global scaling and waveform displays.

When creating a new waveform display page, the minimum and maximum values for each line are found and stored on the disk with each data line's starting time code. In local scaling, each line is scaled to its own minimum and maximum, while, page scaling uses the page minimum and maximum to determine the scaling for each line. The parameter, ISWIT, is used in global scaling to enable WAVE to retrieve all the page data and determine the page minimum and maximum before any lines of data are displayed.

Teh details option enables a user to zoom in and expand across the display screen. The minimum and maximum values for each line may change with this option, therefore, the minimum and maximum values have to be determined each time a page is displayed or re-displayed. Details utilizes the parameters, ITH and JTH, which are the indices for the first and last power terms in the frequency band of interest. If ITH equals 1 and JTH equals NPTS, the Details option is off and the complete spectrum will be displayed.

Calling Sequence: CALL WAVE

Subroutine Parameters: None

Common Area: Same as EXEC

File: TEXT

Program Name: POW

Function: POW puts, in text form, all the current status of IDRS at the top of the display page. This status tells what switches are active and what values are associated. POW also prepares some parameters for the power overlay.

Calling Sequence: CALL POW

Common Area: Same as EXEC

File: TEXT

Program Name: WAV

Function: WAV puts, in text form, all the current status of IDRS at the top of the display page. This status tells what switches are active and what values are associated. POW also prepares some parameters for the power overlay.

Calling Sequence: CALL WAV

Subroutine Parameters: None

Common Area: Same as EXEC

File: TEXT

Program Name: FILTXT

Function: This routine prints filter parameters for wave or power spectrum. Filter parameters are expected to be already in text form in ISTEXT (7-15).

Calling Sequence: CALL FILTXT(II)

Subroutine Parameters:

II (Integer) = 1; Filter 1
2; Filter 2

Common Area: Unlabled resident

File: HIST2

Program Name: EXECUTIVE

Function: Computes and displays a two-dimensional histogram of one or two channel data.

Calling Sequence: None

Common Area: None

File: HIST2

Program Name: HISD2

Function: Displays one line of a two-dimensional histogram.

Calling Sequence: CALL HISD2(NX, J, H, NY, ISTAT, ICH)

Subroutine Parameters:

- NX - Number of bins in the x-direction
- J - The number of the line to be displayed
- H - The array of length NX that contains the histogram data of the JTH line
- NY - The total number of lines in the display
- ISTAT - Run number
- ICH - Channel number

Common Area: None

File: HIST2

Program Name: GETLIN

Function: Reads and stores one line of histogram data.

Calling Sequence: CALL GETLIN(NB, I, HS, ILS)

Subroutine Parameters:

- NB - Length of the line of data
- I - The number of the line to be read and stored
- HS - The array into which the line of data will be read
- ILS - The number of the last sector of data

Common Area: None

APPENDIX A

Approach To Interactive Digital Filter Design Capability

A.1. INTRODUCTION

There was a need in IDRS to be able interactively to design a digital filter with prescribed pass band characteristics (low pass, high pass, band pass, or band elimination). The user would also like to have some control on attenuation roll-off. The filters available in IDRS are Chebyshev and Butterworth type, where the roll-off is indirectly prescribed through the number of poles in the filter. The approach taken in designing these filters in IDRS is discussed in this appendix.

A.2. DESIGN PHILOSOPHY

A filter is one of the very important components in any signal processing system - digital or analog - to filter noise or unwanted signals, to tune in desired signal channels, etc. Quite a large number of techniques have become available over the past few years to design many different types of digital filters. Some techniques are more complex than others; for example, some of them may require the solution of a system of non-linear equations or the optimization of the filter parameters as part of the design process. It is a common experience that prediction of the convergence of such solutions is a very difficult problem. Such techniques normally require extensive interaction and therefore a good theoretical background of the network on the part of the designer. However, there are certain designs for which the solutions are available in closed form or always guaranteed. A computer program could be generated for such designs that would not require any theoretical background on the part of its user. Of course, the filters obtained by different design techniques will have different characteristics, but the simplicity and straightforwardness of the design can be important criteria. It is more than likely that a normal user of IDRS (Interactive Digital Receiver Simulator) will have little knowledge of or interest in knowing the details of the filter design techniques. All his interest will be is in using the digital filter. Such considerations weigh heavily in favor of using those design techniques for which solutions are guaranteed in advance. The advantages and disadvantages of such filters will be detailed later in this presentation.

A.3. DIGITAL FILTER CLASSIFICATION

The digital filters are mostly classified according to their impulse response characteristics (1) i.e.,

(1) Infinite Impulse Response (IIR)
(2) Finite Impulse Response (FIR)

```
                                                      FILTER DESIGN AND DISPLAY FRAME
WANT NEW DESIGN OR DISPLAY CURRENT FILTER?
0 FOR NEW DESIGN
1 FOR DISPLAYING CURRENT  FILTER CHARAVTERISTICO
FILTER CLASS:
0 FOR CHEBYSHEV FILTER
1 FOR BUTTERWORTH FILTER  0
PASS -BAND CHARECTERISTICS:
0 FOR LOW-PASS,
1 FOR HIGH PASS,
2 FOR BAND PASS, AND
3 FOR BAND ELIMINATION:0
SPECIFY CUT OFF FREQUENCIES
TYPE UPPER CUT OFF FREQUENCY: 25
TYPE DESIRED NUMBER OF POLES(MAXIMUM ALLOWED IS 18):    6
     NPOLE1=    GNFLGPB=    0NWCL,NWCH=    0   25
  6    0    0  0.0000000E 00  0.5000000E-01  0.0000000E 00
       ZB1            ZB2            ZB3            ZB4            AK

  0.2000000E 01  0.9999999E 00  0.0000000E 00  0.0000000E 00  0.2379315E-01

 0.2000000E 01  0.9999999E 00  0.0000000E 00  0.0000000E 00  0.1310235E-01

 0.2000000E 01  0.9999999E 00  0.0000000E 00  0.0000000E 00  0.2905656E-02

       ZA1            ZA2            ZA3            ZA4

  0.1867114E 01  -0.9622863E 00  0.0000000E 00  0.0000000E 00

 0.1846799E 01  -0.8992079E 00  0.0000000E 00  0.0000000E 00

 0.1851823E 01  -0.8634453E 00  0.0000000E 00  0.0000000E 00

    AK(I)=  0.2379315E-01  0.1310235E-01  0.2905656E-02
1  FOR  ANOTHER  DESIGN
2  FOR  DISPLAYING CHARACTERISTICS
3  FOR  EXIT  FROM  THIS  FRAME
```

Figure A-1 A Typical Question and Answer Session for Filter Design

A finite impulse response filter is one whose impulse response is zero outside some finite limits. An infinite impulse response filter is a filter whose impulse response is an infinite sequence in the time domain.[1] One of the most important features of FIR filters is that they have perfectly linear phase characteristics. The FIR filters have been found to be more complicated to design. Also, the execution time of FIR filters (number of arithmetic multiplications) has been found to be greater than that of IIR elliptic filters.(2) The elliptic filters do not have the linear phase characteristics, but the delay equalizers could be designed. However, the equalized elliptic filters would no longer be efficient (in terms of multiplicative operations) as compared to FIR filters.[2]

We have chosen to implement Butterworth and Chebyshev (IIR) filters in LIPS due to their mathematical simplicity, which is in accordance with our design philosophy stated earlier. A user can interactively design an up to 20-pole Butterworth or Chebyshev filter with low pass, band pass, band elimination, or high pass characteristics. The upper limit of 20 poles is arbitrary and could be easily changed by changing the appropriate dimension statements in the FORTRAN programs. The input parameters needed for the design are:

(1) Filter Type
(2) Number of Poles
(3) Pass Band Characteristics
(4) Cut-Off Frequencies

A typical question and answer session for inputing these parameters is shown in Figure A-1.

A.4. DESIGN PROCESS

Once the filter parameters have been specified, the design process starts with choosing the transfer function of an appropriate analog filter H(S) and the formation of a digital transfer function H(Z) from this analog function through some kind of transformation. This is quite a reasonable approach, as the art of analog filter design is highly advanced, with many relatively simple closed-form design formulas. It will be simple to implement digital filter designs based on such formulas.

In transforming analog function H(a) to digital function H(Z) we require that the essential properties of the analog frequency response be preserved in the frequency response of the resulting filter. There are two main techniques available for this transformation known as [3]

(1) Impulse Invariance and
(2) Bilinear Transform

In the "Impulse Invariance" method, the unit sample response of the digital filter h(t) is chosen as the equally-spaced samples of the impulse response of the analog filter $h_a(t)$; i.e.,

$$h(n) = h_a(nT)$$

where T is the sampling period.

It is a well-known property of the sampling process that the $H_a(S)$ and $H(Z)$ will be equal only if $H_a(S)$ is band limited. Otherwise, they will be different due to a phenomenon known as "aliasing." Unfortunately, any practical analog filter will not be band limited; consequently, the frequency response of the digital filter designed by the "Impulse Invariance" technique will in fact be different from the response of the prototype analog filter.

The second technique, "Bilinear Transform", is the one that is used in the present filter package for converting a prototype analog filter to a digital filter. The analog and digital frequency variables S and Z in this technique are related as:

$$S = \frac{2}{T} \frac{1-Z^{-1}}{1+Z^{-1}} \qquad (1)$$

It can be seen from this relation that the frequency axis in S-domain ($S=j\Omega$) is mapped onto the unit circle in Z domain ($Z=e^{j\omega}$) as

$$\frac{T\,\Omega}{2} = \tan(\omega/2) \qquad (2)$$

The positive and negative imaginary axes of the S-plane are mapped, respectively, into upper and lower halves of the unit circle in the Z-plane as shown in Figure A-2. The Bilinear Transformation avoids the problem of aliasing encountered with impulse invariance; however, it introduces a distortion in the frequency axis. This is not a problem for the class of filters to be designed presently, since we are interested in designs that are constants in a certain band of frequency (LP, BP, HP, or BE) and the distortion of the frequency axis will not change the amplitude characteristics.

A.5. FLOW CHART - DESIGN ALGORITHM

The flow graph of the design algorithm is shown in Figure A-3. An appropriate low pass Chebyshev or Butterworth function with a normalized low pass cut-off frequency of unity is chosen as:

$$|H(S)|^2 = \frac{1}{1+\left(\frac{S}{j}\right)^{2N}} \qquad (3)$$

for Butterworth filters and

$$|H(S)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(S/j)} \qquad (4)$$

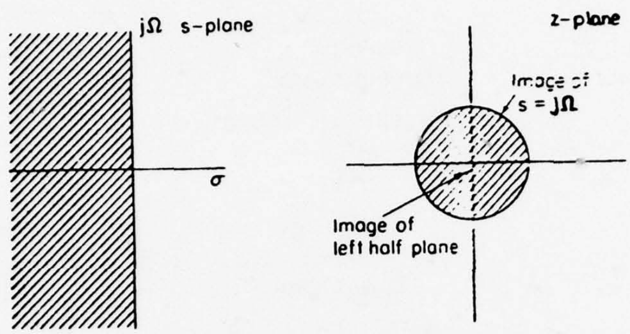


Figure A-2(a) Mapping of the s-plane into the z-plane using the bilinear transformation

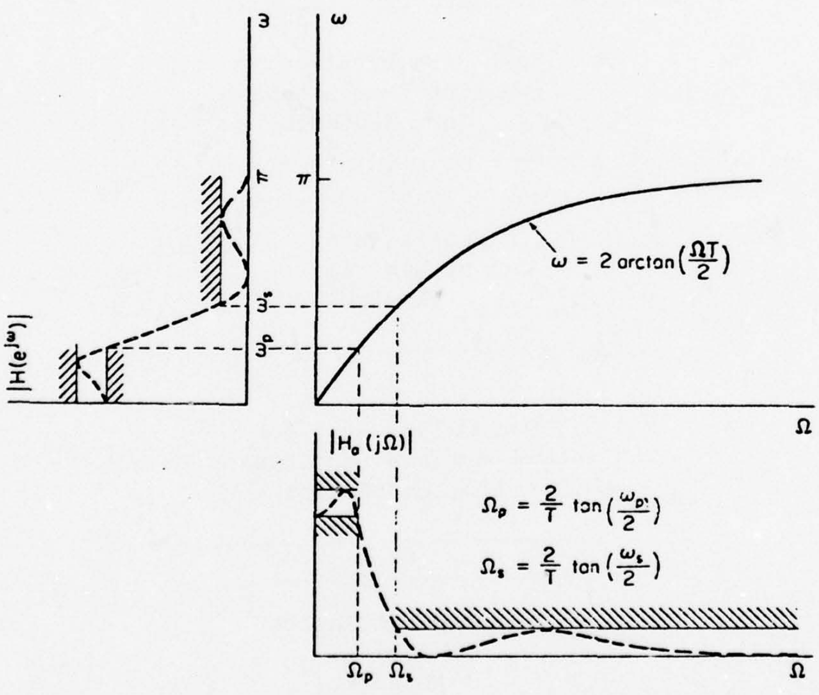


Figure A-2(b) Frequency warping encountered in transforming an analog low pass filter to a digital low pass filter. To achieve the desired digital cutoff frequency, the analog cutoff frequencies must be prewarped as indicated.

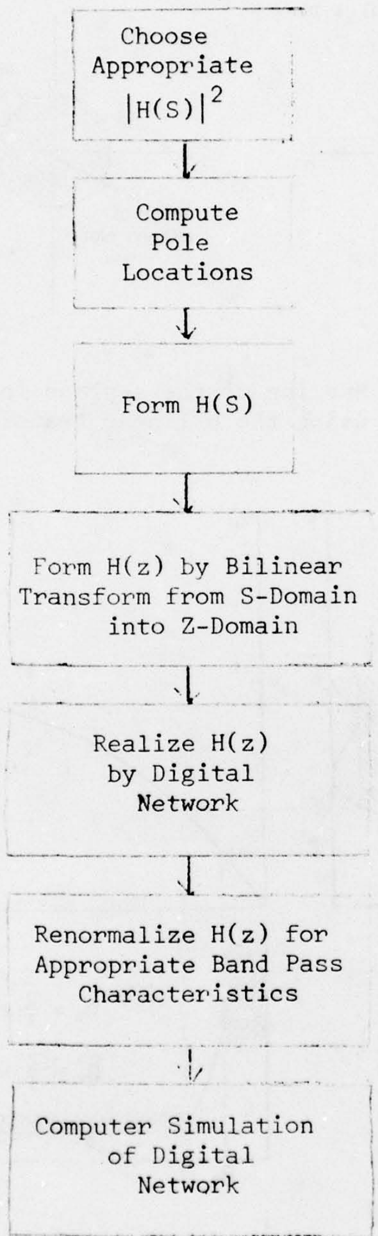


Figure A-3 Block Diagram: Design Algorithm

for Chebyshev filters. The variable

S (= $\sigma + \Omega j$) is the complex analog frequency variable,

N is the number of poles,

and ε is the pass band ripple width of the filter.

$T_n(X)$ in equation (4) is the N^{th} order Chebyshev polynomial.

The transfer function $H(S)$ is formed from the left half plane (LHP) poles of the above $|H(S)|^2$. These poles, shown in Figure A-4, are obtainable in closed form as (4)

$$S = -\sin \frac{(2\gamma+1)\pi}{2N} + j \cos \frac{(2\gamma+1)\pi}{2N}$$

for Butterworth filters and

=0,1,2,...n-1

$$S = -\sinh \phi_2 \sin \frac{(2\gamma+1)\pi}{2N} + j\cosh \phi_2 \cos \frac{(2\gamma+1)\pi}{2n} \qquad (5)$$

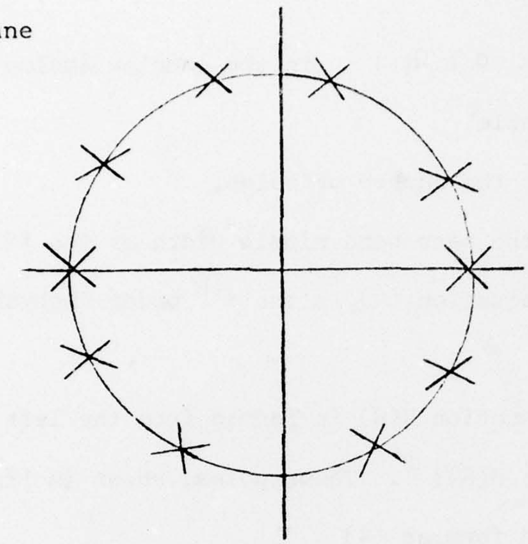
for Chebyshev filters.

$$\text{where } \phi_2 = \frac{1}{N} \sinh^{-1} \frac{1}{\varepsilon} \qquad (6)$$

Thus $H(S)$ can be obtained from LHP poles as

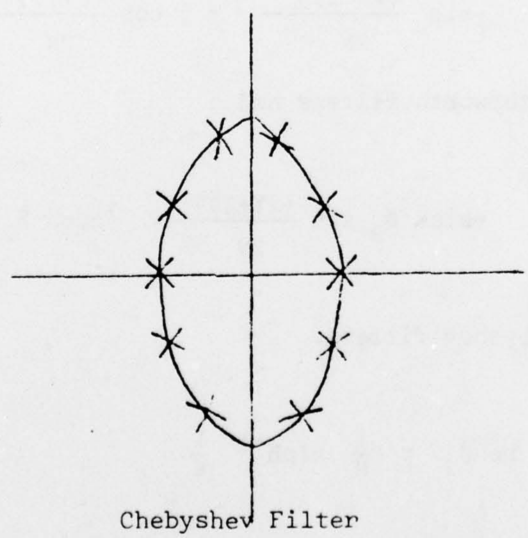
$$H(S) = \frac{K}{(S^2+a_{11}S+a_{21})(S^2+a_{12}S+a_{22})\ldots(S^2+a_{1i}S+a_{2i})\ldots(S+a)} \qquad (7)$$

S-Plane



Butterworth Filter

S-Plane



Chebyshev Filter

Figure A-4 Pole Locations in S-Plane

and then H(Z) can be obtained through the transform of equation as

$$H(Z) = \frac{A(1+Z^{-1})^n}{(1-\alpha_{11}Z^{-1}-\alpha_{21}Z^{-2})\,(1-\alpha_{12}Z^{-1}-\alpha_{22}Z^{-2})\ldots(1-\alpha_{1i}Z^{-1}-\alpha_{2i}Z^{-2})\ldots \ldots(1\alpha_{in}Z^{-1})} \qquad (8)$$

A.6. NETWORK REALIZATION OF H(Z)

There are many different network structures (shown in Figure A-5) available for realizing the above H(Z), some of them more straightforward to synthesize than others (Ref. 5). The simplest of them, the direct form realization, is not canonic in the sense that the number of storage elements required by this structure is not minimal. Therefore this structure is rarely used. Some Canonic structures are also shown in Figure A-5. A natural question arises, "Is any one of them superior to the other and, if so, in what respect?"

Digital filtering is realized by adders, multipliers, and delay elements as opposed to the resistors, capacitors, and inductors of analog filters. The number of bits (word length) used in realizing these adders, multipliers, etc., is finite only. The cost of the digital hardware increases exponentially with the word length, and therefore we would like to keep this word length as small as possible. However, the truncation of the intermediate multiplications and additions due to smaller word length introduces errors caused by finite register length and quantization noises. Considerable research has been done in recent years to develop digital structures that are less sensitive to such effects. The cascade form of realization (Figure A-5) has been found to be less sensitive to these effects as compared to the other structures shown in the same figure. This is not to say that the cascade form is the least sensitive. There are other more complex network structures that are much less sensitive than the cascade form and where only 8 bits (or sometimes even 4 bits) of arithmetic is good enough to give adequate performance. Such structures are more complex to realize. Since we are going to simulate these digital filters on the the PDP-11, which has floating point hardware, the effect of finite register length is almost negligible. As a matter of fact, the performance of any new digital structure is usually compared by simulating the structure both in floating point arithmetic and in lower-bit fixed point arithmetic, taking the response of simulation in floating point as standard. Thus the finite word length is not expected to be a

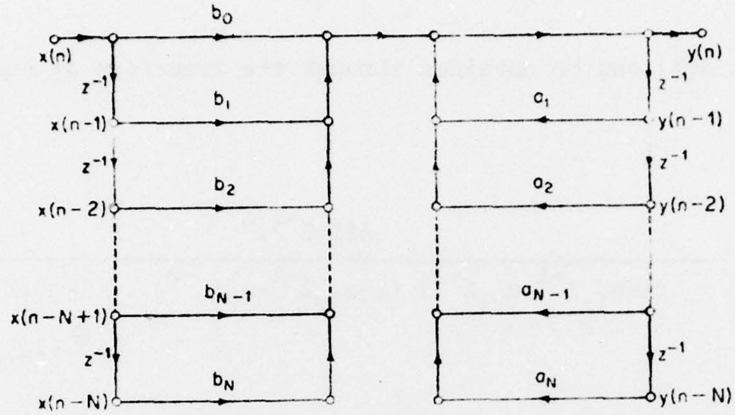


Figure A-5(a) Direct Form Realization

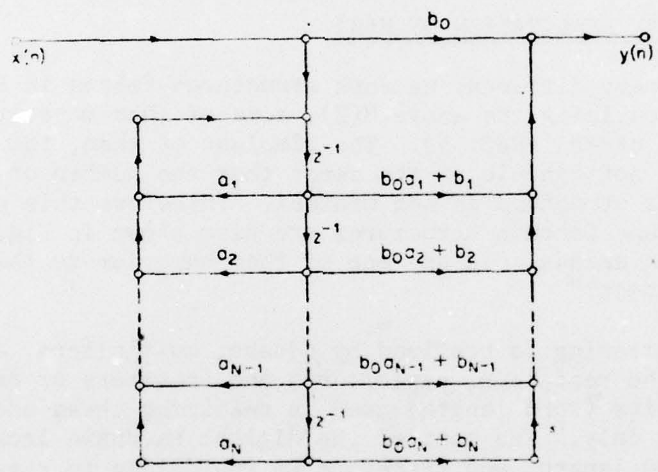


Figure A-5(b) Canonic Realization

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}$$

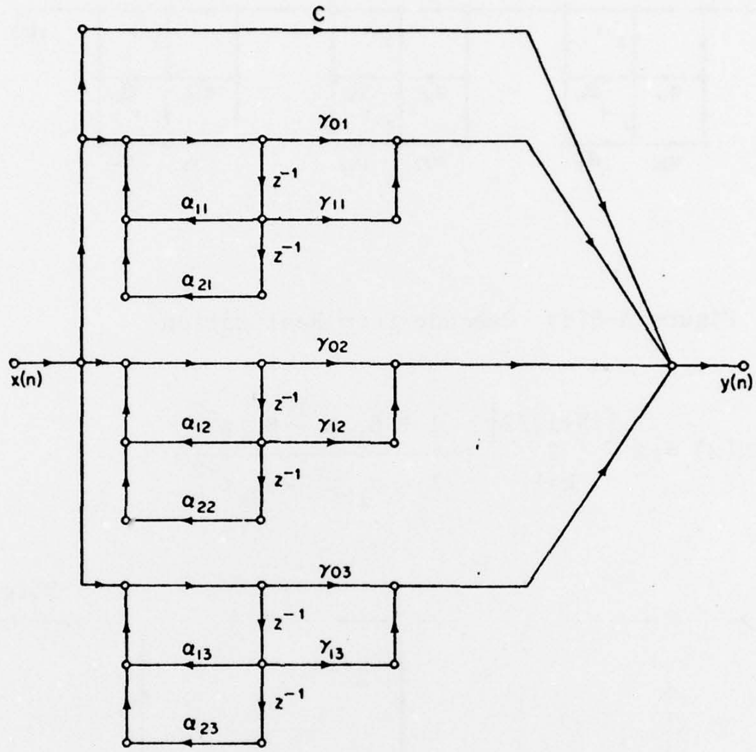


Figure A-5(c) Parallel Form Realization

$$H(z) = \sum_{k=0}^{M-N} C_k z^{-k} + \sum_{k=1}^{\lfloor (N+1)/2 \rfloor} \frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}}$$

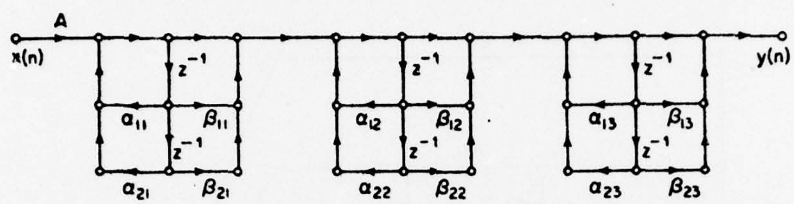
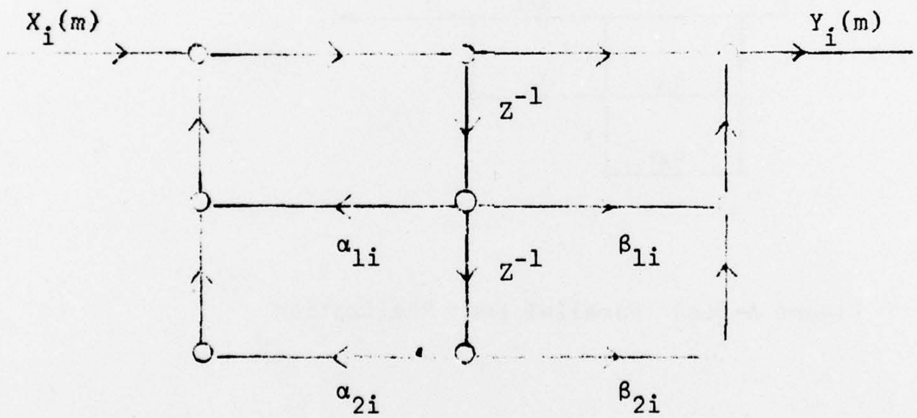


Figure A-5(d) Cascade Form Realization

$$H(z) = A \prod_{k=1}^{\left[(N+1)/2\right]} \frac{1 + \beta_{1k}z^{-1}+\beta_{2k}z^{-2}}{1 - \alpha_{1k}z^{-1}-\alpha_{2k}z^{-2}}$$



$$H(Z) = \frac{1 + \beta_{1i}Z^{-1} + \beta_{2i}Z^{-2}}{1 - \alpha_{1i}Z^{-1} - \alpha_{2i}Z^{-2}}$$

Figure A-5(e) Realization of i^{th} - Biquadratic Section

problem in the present IDRS system. Since it is relatively more straightforward to synthesize a given H(Z) in cascade form than in other forms, we have chosen the cascade form for implementation in our present system.

The cascade form is realized by arranging the transfer function H(Z) (equations 7 or 8) as the product of biquadratic factors

$$H(Z) = \prod_{k=1}^{[(N+1)/2]} \frac{1 + \beta_{1K} Z^{-1} + \beta_{2K} Z^{-2}}{1 - \alpha_{1K} Z^{-1} - \alpha_{2K} Z^{-2}} \qquad (9)$$

This arrangement of H(Z) directly gives the multiplier coefficients of cascade structure shown in Figure A-5. Equation (9) is quite a general expression and some of these coefficients in some sections may be zero. In a software implementation of this structure, a subroutine 'BIQUAD' is programmed to simulate one section of the cascade (Figure A-5(e)) with coefficients α_1, α_2, β_1, β_2, $X_i(n)$ and $Y_i(n)$ as the input parameters. This subroutine is called repeatedly for different sections to compute the final output of the digital filter.

A.7. FREQUENCY-BAND TRANSFORMATION

The digital filter's response is essentially periodic, with the sampling frequency as its period (see Figure A-6). The different ways in which the cut-off frequency w of a digital filter can be specified are:

(i) w in degrees (with fold-over frequency as 180° or sampling frequency as 360°).

(ii) w as a fraction of sampling frequency or as a fraction of fold-over frequency.

(iii) The actual cut-off frequency and the sampling rate.

All the different methods of specification are equivalent and any one could be used. It shoud be remembered, however, that ultimately, digital filter design depends only upon the ratio of cut-off frequency and sampling frequency. Thus, for example, the same filter could be used for a system where both sampling rate and cut-off frequency are twice those of the original design. Since the sampling rate is a global parameter in IDRS system, it is convenient for the operator to specify the actual cut-off frequency as compared to computing it as a sampling frequency fraction beforehand. Once the cut-off frequency has been specified, the analog frequency is obtained as:

$$\Omega_c = 2 \tan(\pi fc/fs) = 2 \tan(\theta p_{/2})$$

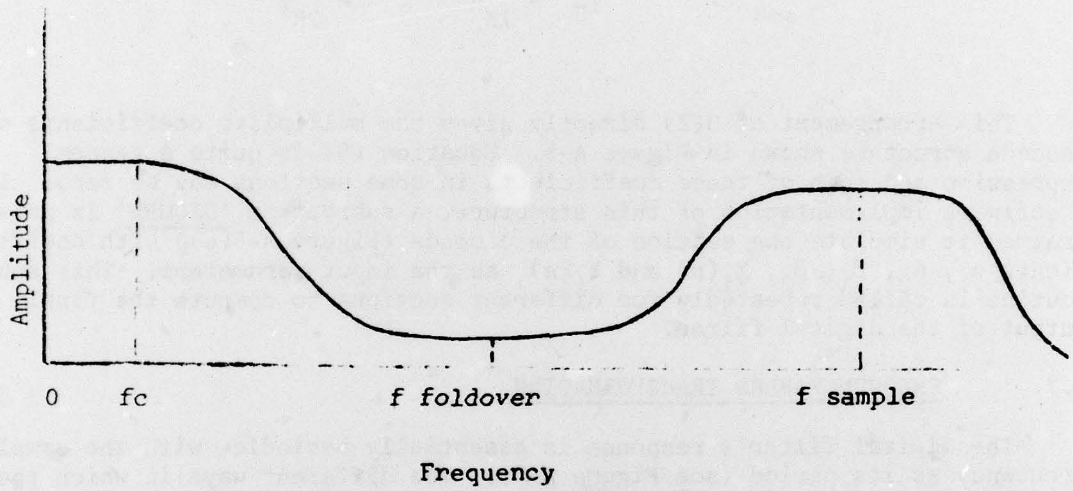


Figure A-6 Frequency Response of a Low Pass Digital Filter

where fs is the sampling frequency. From the computer programming point of view, it is simpler to determine the analog transfer function for unity cut-off frequency ($\Omega_c = 1$). The only digital filter designed in this manner, so far, is of low-pass nature with cut-off frequency θ_p (=0.927295 radians). However, it can be transformed to any other pass band characteristic by the use of appropriate spectral transformations. These transformations, first published in 1970 (ref. 6) are reproduced here in Figure A-7 for easy reference.

A.8. SUMMARY

To summarize, the IDRS system can interactively design a 18-pole Butterworth or Chebyshev digital filter. The user can select low pass, high pass, band pass, or band elimination characteristics. A typical question and answer session for filter design is shown in Figure A-1. This designed filter could then be used to filter any waveform in the receiver section of IDRS.

Filter Type	*Transformation*	*Associated Design Formulas*
Lowpass	$z^{-1} = \dfrac{Z^{-1} - \alpha}{1 - \alpha Z^{-1}}$	$\alpha = \dfrac{\sin\left(\dfrac{\theta_p - \omega_p}{2}\right)}{\sin\left(\dfrac{\theta_p + \omega_p}{2}\right)}$ ω_p = desired cutoff frequency
Highpass	$-\dfrac{Z^{-1} + \alpha}{1 + \alpha Z^{-1}}$	$\alpha = -\dfrac{\cos\left(\dfrac{\omega_p + \theta_p}{2}\right)}{\cos\left(\dfrac{\omega_p - \theta_p}{2}\right)}$ ω_p = desired cutoff frequency
Bandpass	$-\dfrac{Z^{-2} - \dfrac{2\alpha k}{k+1} Z^{-1} + \dfrac{k-1}{k+1}}{\dfrac{k-1}{k+1} Z^{-2} - \dfrac{2\alpha k}{k+1} Z^{-1} + 1}$	$\alpha = \dfrac{\cos\left(\dfrac{\omega_2 + \omega_1}{2}\right)}{\cos\left(\dfrac{\omega_2 - \omega_1}{2}\right)}$ $k = \cot\left(\dfrac{\omega_2 - \omega_1}{2}\right) \tan \dfrac{\theta_p}{2}$ ω_2, ω_1 = desired upper and lower cutoff frequencies
Bandstop	$\dfrac{Z^{-2} - \dfrac{2\alpha}{1+k} Z^{-1} + \dfrac{1-k}{1+k}}{\dfrac{1-k}{1+k} Z^{-2} - \dfrac{2\alpha}{1+k} Z^{-1} + 1}$	$\alpha = \dfrac{\cos\left(\dfrac{\omega_2 + \omega_1}{2}\right)}{\cos\left(\dfrac{\omega_2 - \omega_1}{2}\right)}$ $k = \tan\left(\dfrac{\omega_2 - \omega_1}{2}\right) \tan \dfrac{\theta_p}{2}$ ω_2, ω_1 = desired upper and lower cutoff frequencies

Figure A-7 Table of Transformations from a Lowpass-Digital-Filter Prototype of Cutoff Frequency θ_p *

* From Oppenheim & Schafer (Ref 3)

REFERENCES

1. Rabiner et al., "Terminology in Digital Signal Processing," IEEE Transactions on Audio and Electro Acoustics, Vol. AU-20, No. 5, Dec 1972, pp. 332-337.

2. Rabiner, Kaiser, Herrmann, and Dolan, "Some Comparisons Between FIR and IIR Digital Filters," Bell System Technical Journal, Vol 53 #2, Feb 1974, pp. 305-331.

3. Oppenheim and Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

4. Weinberg, Network Analysis and Synthesis, McGraw Hill Book Company, Inc., New York, 1962.

5. A.G. Constantinides, "Frequency Transformations for Digital Filters," Proc. IEEE, Vol 117, No. 8, Aug 1970, pp. 1585-1590.

APPENDIX B

EIP VERSION OF IDRS

The EIP version of IDRS represents several modifications and additions to the baseline LWS/IDRS package operating in a DEC-DOS (disk operating system) environment. The time waveform segmentation and line-oriented signal processing-display philosophy of the original IDRS software was combined with the feature of a new overlay supporting a directory file structure on the RP04 disk pack. These features allow on-line interactive multiple waveform storage and processing utilizing the basic IDRS routines. During the initial design phase of LWS/IDRS, a program philosophy and data structure was sought which would allow flexible signal processing of waveforms having arbitrary length. This design condition was satisfied by structuring the program such that the signal processing and storage scheme allowed one input waveform to utilize as much as 20 percent of the RP04 disk space. Although this structure supported extremely long waveforms, only one waveform could be resident on the disk during program execution. A need for on-line storage and processing of many distinct waveforms was encountered during the data analysis phase of the EIP Verona data collection effort. A typical data analysis task for the above effort may consist of visual inspection or demodulation of hundreds of emitter waveforms during one processing session. The basic LWS/IDRS package could not efficiently accommodate this processing task. The EIP version of IDRS was modified for this purpose.

B.1. CONFIGURATION

The EIP version of IDRS requires the same minimal hardware configuration as the original system operating under the RSX system. However, proper use of the directory option for the multiple waveform processing options requires an appropriately formatted RP04 disk pack.

B.2. MODIFICATIONS

The directory (DIR) overlay extends the use of IDRS to the processing of multiple waveforms either in a sequential mode or parallel fashion. This is accomplished by allowing the user to change or append to the current IDRS data set (i.e., waveforms under IDRS control). When the DIR overlay is core resident, the user may execute commands to transfer a set of waveforms from the EIP data base to IDRS working space and perform other directory manipulation tasks.

A list of the user commands available with EIP version of IDRS is given below:

- /L - list entire directory
- /D - delete file
- /G - retrieve EIP file and transfer to IDRS working space

- o /T - write WPS compatible tape of specified file
- o /S - write WPS compatible tape of current data set
- o /E - return to IDRS executive

Due to the short time duration and large time-bandwidth product nature of the EIP signals, several modifications to the IDRS synchronous detector were necessary. The major modification concerned the addition of an RF preconditioning filter and a low-pass postdetection filter in the synchronous detection chain. All the filtering is accomplished in one run through the IDRS synchronous detector. The filters are independently designed and stored on the RP04 disk by a modified version of the FILDSG overlay. Operating procedure concerning these modifications is self-explanatory at time of execution. Other minor modifications to the baseline IDRS are evident from the MENU list of Figure B-1. The purpose of these minor changes is indicated in the figure.

P-RESET PARAMETERS
 RL-RESET # LINES
 RP-RESET # PTS/LINE
 RE-RESET SKIP POINTS (RESAMPLING)
 RO-RESET OVERLAP
 RS-RESET SKIP POINTS

M-MENU

LL-LOCAL LINE SCALE
LP-LOCAL PAGE SCALE
GW-GLOBAL WAVE SCALE

W-DISPLAY WAVE
P-DISPLAY POWER SPECTRUM
S-DISPLAY SAME DATA

X-TEXT (PAGE)

T-INPUT NEW TIME
T+-ADVANCE AHEAD TIME
T--ADVANCE BACK TIME

N OR NP-NEXT PAGE
WN-WAVEFORM NEXT
PN-POWER NEXT

L--PREVIOUS LINE (BACK 1 LINE)
L+-NEXT LINE (AHEAD 1 LINE)

DP-DETAILS POINTS (ITH,JTH)
DL-DETAILS LINE (EXPLODE 1 LINE)
DO-DETAILS OFF

DI-DATA ON DISK
TI-TAPE ROLLIN
TO-TAPE ROLLOUT
TD-TAPE DUMP (PAGE)
TW-WPS FORMATTED TAPE
TP-TAPE PLAYBACK (PAGE)

PA-POWER SPECTRUM AVERAGE
PL-LOG OF POWER PP-POWER PROFILE
PH-HANNING/HAMMING WEIGHTING

DM-DEMODULATION ON
FQ-FREQ. CONV.
DQ-PHASE/DT
FI-FILTER INIT
FN-FILTER NO INIT
WR-WAVEFORM RECTIFICATION

MO-DEMODULATION OFF
QO-FREQ. CONV. OFF
HO-WEIGHTING OFF
WO-WAVE OPTIONS OFF
PO-POWER OPTIONS OFF
FO-FILTER OFF

SO-STORE MODE OFF
O-ALL OPTIONS OFF

SW-STORE MODE (WAVE)
SP-STORE MODE (POWER)

D1-DISPLAY MODE 1
D2-DISPLAY MODE 2

FD-FILTER DESIGN

WL- WAVEFORM LOG

WD- WINDOW

SQ-SQUARE WAVEFORM

DR-DIRECTORY AND FETCH

SR-SAMPLING RATE INPUT

K-KILL RETURN TO MONITOR

DD-DOUBLE DISPLAY MODE I/Q,AM/FM
 SD-SINGLE DISPLAY MODE

ST STORE CURRENT PAGE ON RP-04

FX-FEATURE EXTRACTION N POINT DFT
RT-RETRANSFORM (ALL OFF - STORE OFF)

Figure B-1 IDRS Menu (EIP Version)